

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Manažment príspevkov počas priebehu konferencie
v reálnom čase

Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Manažment príspevkov počas priebehu konferencie
v reálnom čase

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: RNDr. Marek Nagy, PhD.

Bratislava, 2020

Jozef Mizerik

Čestné vyhlásenie

Čestne prehlasujem, že som bakalársku prácu s názvom „Manažment príspevkov počas priebehu konferencie v reálnom čase“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, s použitím uvedených zdrojov.

V Bratislave, 01.06.2020

Jozef Mizerík



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Jozef Mizerík
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Manažment príspevkov počas priebehu konferencie v reálnom čase
Real-time management of contributions during the conference

Anotácia: Priebeh konferencie je náročný na manažment. Je potrebné prezentované príspevky naplánovať do rôznych sekcií a prezentačných miestností. Samotný priebeh konferencie prináša nečakané zmeny, ktoré treba zohľadniť.

Cieľ: Vytvoriť pomocnú real-time webovú aplikáciu, ktorá bude zohľadňovať nasledujúce fázy konferencie:

- 1) Rozdelenie príspevkov do sekcií a miestností. Ich časovanie. Zmeny sa vykonávajú v reálnom čase a premietnu sa do zobrazeného programu konferencie.
- 2) Podpora priebehu konferencie. Vedúci sekcie bude mať privátny prístup (napr. cez tablet), kde bude posúvať priebeh príspevkov. Bude mať spätnú väzbu o časovaní a synchronizácii s ostatnými sekciami, aby usmernil diskusiu. Súčasťou systému budú aj časovacie zariadenia (napr. tablety), ktoré zobrazujú čas prednášajúcemu. Nastavuje ich vedúci sekcie.

Použije sa JavaScript, socket.io, node.js.

Vedúci: RNDr. Marek Nagy, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 19.09.2019

Dátum schválenia: 07.10.2019

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie

Rád by som poďakoval vedúcemu práce RNDr. Marekovi Nagyovi, PhD. za nadštandardnú ochotu, odborné rady a praktické pripomienky pri vypracovávaní tejto práce.

Jozef Mizerík

ABSTRAKT

Manažovanie konferencie je náročné. Konferencia môže trvať niekoľko dní, uskutočniť sa vo viacerých miestnostiach a v každej z týchto miestností prednášajú svoj príspevok mnohí prednášajúci. A netreba zabúdať na účastníkov konferencie, z ktorých každý sa chce zúčastniť príspevkov, ktoré si vyberie a žiadny z nich nezmeškať. V rámci práce sme vytvorili systém na manažovanie priebehu konferencie v reálnom čase podľa požiadaviek zadávateľa. Tento systém rieši problémy, ku ktorým doteraz pri organizovaní konferencie dochádzalo, ako napríklad nečakané zmeny v rozvrhu alebo meškania jednotlivých príspevkov. Navyše poskytuje organizátorom veľké množstvo podpornej funkcionality, čo zjednoduší a urýchli ich prácu. Rovnako pomáha účastníkom konferencie, ktorí vidia v reálnom čase, ktorý príspevok práve prebieha a koľko času zostáva do jeho konca. Lepšie si tak môžu naplánovať, kedy sa do ktorej miestnosti presunúť. Pri tom všetkom si systém zachováva jednoduchosť a intuitívnosť.

Kľúčové slová: konferencia, v reálnom čase, webová aplikácia, JavaScript

ABSTRACT

Conference management is challenging. A conference could last couple of days, take place in several rooms and a lot of speakers have their presentations in each of these rooms. We should not forget conference participants who want to participate in contributions which they have chosen. They do not want to miss any of them. In the thesis, we created a system for conference management in real time. The system is designed in accordance with product requirements. The system solves problems which have occurred during conferences such as unexpected changes in the schedule or delays in individual contributions. In addition, it provides organizers with a large amount of support functionality, which simplifies and speeds up their work. It is also helpful for conference participants who see in real time which contribution is in progress and how much time remains until its end. This way, they can better plan when to move to which room. At the same time, the system retains its simplicity and intuitiveness.

Key words: conference, real-time, web application, JavaScript

Obsah

Úvod.....	1
1 Východiská	3
1.1 Špecifikácia požiadaviek	3
1.1.1 Predmet špecifikácie	3
1.1.2 Slovníček pojmov	3
1.1.3 Priebeh konferencie	4
1.1.4 Problémy počas priebehu konferencie	5
1.1.5 Charakteristika programu.....	6
1.1.6 Identifikované roly.....	6
1.2 Prehľad existujúcich systémov: MeetingHand	7
1.2.1 Vytvorenie udalosti.....	7
1.2.2 Prehľad organizátora o konferenciách.....	8
1.2.3 Vytváranie rozvrhu	10
1.2.4 Webstránka z pohľadu návštevníka	11
2 Analýza požiadaviek a návrh systému	14
2.1 Charakteristika rolí.....	14
2.2 Rola organizačný výbor	14
2.2.1 Úprava rozvrhu	15
2.2.2 Členovia	17
2.2.3 Príspevky.....	18
2.2.4 Nastavenia.....	18
2.2.5 Rozvrh.....	19
2.2.6 Nastavenia účtu	19
2.3 Rola vedúceho sekcie.....	19

2.4 Rola používateľ časovača	20
2.5 Rola návštevníka.....	21
2.6 Modely	21
2.7 Použité technológie a nástroje	23
2.7.1 HTML, CSS, JavaScript	23
2.7.2 Node.js	23
2.7.3 NPM – Node manažér balíčkov	24
2.7.4 MongoDB	25
2.8 Architektúra systému	26
2.9 Server	28
2.9.1 Komunikácia s databázou	28
2.9.2 Autentifikácia a autorizácia	28
2.10 Klient.....	29
2.10.1 Vkladanie dynamického obsahu	29
2.10.2 Podpora naprieč prehliadačmi.....	30
2.10.3 Design aplikácie.....	30
2.10.4 Zariadenia s menšou šírkou obrazovky.....	31
3 Implementácia.....	32
3.1 Export a import modulov	32
3.2 Používateľské rozhranie.....	33
3.3 Zobrazovanie na zariadeniach s menšou obrazovkou.....	36
3.4 Časovače	37
3.5 Zobrazovanie v reálnom čase bez potreby aktualizácie stránky	38
4 Testovanie	40
4.1 Testovacie scenáre pre manažovanie príspevkov a používateľských účtov	40

4.2 Testovacie scenáre pre tvorbu rozvrhu	41
4.3 Testovacie scenáre pre priebeh konferencie v reálnom čase	43
4.4 Testovanie zabezpečenia aplikácie	44
Záver	45
Zdroje	47
Zoznam obrázkov.....	49
Zoznam príloh na CD.....	50

Úvod

Pod pojmom konferencia sa vo všeobecnosti rozumie stretnutie niekoľkých ľudí, ktorí medzi sebou diskutujú o nejakej téme. Experti si medzi sebou vymieňajú nové informácie a často vzniknú nové inovatívne myšlienky.[1] Konferencia môže trvať niekoľko dní, uskutočniť sa vo viacerých miestnostiach, každá miestnosť ešte môže byť rozdelená na niekoľko sekcií a každá sekcia obsahovať niekoľko príspevkov, od mnohých prednášajúcich. Sekcie majú svojich vedúcich, ktorí uvádzajú jednotlivé príspevky. A netreba zabúdať na účastníkov konferencie, z ktorých každý sa chce zúčastniť vybraných príspevkov a žiadny z nich nezmeškať. Manažment toľkých ľudí a činností je náročný.

Predstavme si prípad, že organizátori večer pred konferenciou vytlačia rozvrhy pre všetkých účastníkov. Až ráno sa dozvedia, že jeden z prednášajúcich ochorel, takže nepríde. Ďalší zase bude meškať, pretože nestihol lietadlo. V tom momente môžu organizátori buď vytlačiť nové rozvrhy alebo rozdať rozvrhy s chybnými údajmi a každého účastníka upozorniť na chybu.

Našťastie, dnes už sa do oblasti konferencií, rovnako ako do iných oblastí, dostali informačné technológie. Nepotrebujeme rozvrhy viac tlačiť pokiaľ máme rozvrh zobrazený na webovej stránke. Každý účastník si ho môže jednoducho pozrieť na svojom notebooku, tablete alebo mobile. V prípade zmeny starý údaj nahradíme novým. Na papierovom rozvrhu by nebolo možné vidieť, ktorý príspevok práve prebieha, prípadne ktorý sa predĺžil a trvá dlhšie, ako bolo pôvodne v pláne. S webovou aplikáciou to problém nie je. A to sú len dva z mnohých prípadov, keď informačné technológie dokážu zjednodušiť priebeh konferencie.

Vytvoriť takúto webovú aplikáciu je cieľom tejto bakalárskej práce. Aplikáciu, ktorá zjednoduší priebeh konferencie a ešte umocní zážitok návštevníkov. Poskytne veľké množstvo funkcionality, ale so zachovaním intuitívnosti, jednoduchosti a prehľadnosti.

V prvej kapitole práce predstavíme priebeh konferencie, problémy, ktoré pri organizácii konferencie vznikajú a požiadavky zadávateľa na aplikáciu. Taktiež analyzujeme podobný existujúci systém. V druhej kapitole bližšie analyzujeme požiadavky zadávateľa a následne

vytvoríme vhodný návrh systému. V tretej kapitole navrhnutý systém implementujeme. Napokon v štvrtej kapitole otestujeme funkčnosť systému.

1 Východiská

V tejto kapitole si popíšeme potreby a požiadavky zadávateľa, z ktorých budeme následne vychádzať v celej práci. Tiež si analyzujeme existujúci systém, podobný našej aplikácii.

1.1 Špecifikácia požiadaviek

V tejto podkapitole si popíšeme potreby a požiadavky zadávateľa.

1.1.1 Predmet špecifikácie

Táto špecifikácia popisuje požiadavky na program, ktorého cieľom je manažovanie priebehu konferencie.

1.1.2 Slovníček pojmov

Tento slovníček platí ako pre špecifikáciu, tak aj pre analýzu a návrh programu.

Hlavný admin – hlavný člen organizačného výboru. Pri prvom spustení programu je to jediný existujúci prístup. Ako jediný admin má právo vytvárať prístupy pre ďalších členov organizačného výboru.

Organizačný výbor – jeden alebo viacero ľudí, ktorí majú na starosti organizovanie konferencie.

Používateľ aplikácie – ktokoľvek, kto používa túto aplikáciu.

Prezentujúci – návštevník konferencie, ktorý práve prednáša svoj príspevok ostatným návštevníkom

Prvok rozvrhu – je to bunka rozvrhu konferencie skladajúca sa z dvoch častí, prvá časť uvádza čas začiatku príspevku a druhá časť samotný názov príspevku, resp. informáciu o prestávke.

Rola – definuje aké práva má používateľ aplikácie.

Rozvrh – očakávaný plán priebehu konferencie s uvedenými presnými časmi, miestnosťami, sekciami, vedúcimi a jednotlivými príspevkami konferencie.

Sekcia – menšia jednotka v rámci rozvrhu miestnosti, zvyčajne orientovaná tematicky a majúca vlastného vedúceho

Skupina sekcií – zoskupenie sekcií, ktoré prebiehajú približne v rovnakom čase

Vedúci sekcie – nazývaný aj „chair“, je človek, ktorý má pridelenú sekciu. Uvádza prezentujúcich a dohliada na rozvrh tejto sekcie.

1.1.3 Priebeh konferencie

Organizátor v predstihu zverejní výzvu o konaní konferencie. Zúčastnenci o konferenciu sa zaregistrujú. Tí, ktorí majú záujem aj vystúpiť, zvyčajne priložia dokument so základnými údajmi o ich príspevku, tzv. abstrakt. Na konci obdobia pre prihlásenie má organizátor zoznam členov, ktorí sa konferencie zúčastnia. Tento zoznam je rozdelený na dve časti – účastníci, ktorí budú konferenciu len navštevovať a účastníci, ktorí budú aj prezentovať príspevok.

Následne sa začne plánovať samotný priebeh konferencie. Plánovanie priebehu konferencie má na starosti organizačný výbor. Konferencia môže trvať jeden, ale aj viacero dní. V závislosti od množstva návštevníkov sa konferencia naplánuje buď v jednej alebo vo viacerých miestnostiach. Organizačný výbor plánuje presný rozvrh konferencie – kto, aký príspevok prezentuje, v akej miestnosti a kedy. Potrebné je naplánovať aj časy prestávok, zvyčajne má každá prestávka jedno z nasledujúcich označení: prestávka na kávu, obed, večera.

Snahou organizátorov je, aby boli príspevky spoločného charakteru v rámci rozvrhu za sebou. Takéto príspevky sa plánujú do spoločnej sekcie. V závislosti od počtu príspevkov môže sekcia v rámci miestnosti trvať celý deň. Ak je však počet príspevkov v sekciách menší, je možné, že v rámci jednej miestnosti po skončení jednej sekcie začne ďalšia sekcia. Sekcia má väčšinou svojho vedúceho, niekedy to však neplatí a sekcia môže byť aj bez vedúceho. Vedúcich pre jednotlivé sekcie priradzuje organizačný výbor. Vedúci sekcie uvádza jednotlivé príspevky a tiež kontroluje čas – dohliada, aby nedošlo k veľkej časovej odchýlke oproti pôvodnému plánu. Sekcia začína prvým príspevkom od hlavného pozvaného, ktorý povie základný príspevok

k danej téme. Tento príspevok je zvyčajne dlhší ako ostatné. Nasleduje niekoľko kratších príspevkov od návštevníkov, ktorí sa na príspevok k danej téme prihlásili.

Každý návštevník si na základe rozvrhu konferencie naplánuje príspevky, ktorých sa chce zúčastniť. Aj keď sa organizátori snažia rozvrh detailne naplánuvať, často dôjde k nepredvídateľným okolnostiam a jeden, alebo aj viacero pozvaných sa nemôže na konferenciu dostať. O tejto zmene sa zvyčajne dozvie organizačný výbor alebo vedúci sekcie prostredníctvom telefónu alebo e-mailu. V takomto prípade je potrebné ostatných návštevníkov konferencie upozorniť na zmenu. Doteraz sa návštevníci o nečakanej zmene rozvrhu dozvedeli len ústnou formou, čo však prinášalo so sebou problémy najmä, keď sa konferencia konala vo viacerých miestnostiach.

Často sa tiež stáva, že príspevky sekcie meškajú a návštevník konferencie by už mal podľa jeho plánu byť v inej sekcii, v inej miestnosti. V prípade, že sa rozhodne pre odchod do inej miestnosti, môže sa stať, že až tu zistí, že aj príspevky druhej sekcie meškajú. V takom prípade ešte mohol ostať v predchádzajúcej miestnosti.

1.1.4 Problémy počas priebehu konferencie

Ako už bolo spomenuté v predchádzajúcej podkapitole, v súčasnosti pri organizácii konferencií dochádza k niekoľkým problémom. Tieto problémy by mal vyriešiť program, manažujúci priebeh konferencie.

Prvým problémom je samotná tvorba rozvrhu, ktorá zaberie dlhý čas. Program by mal tvorbu výrazne zjednodušiť a urýchliť. Program taktiež poskytne podporu pri hľadaní chýb. Pri výskyte určitých druhov chýb, dokáže používateľa upozorniť, prípadne chyby sám vyriešiť.

Veľkým problémom je neaktuálnosť rozvrhu – nepredvídateľné zmeny v rozvrhu sa často stávajú. Organizátori musia častokrát ráno pred konaním konferencie vytlačiť a rozdať nové rozvrhy, pretože došlo k zmenám. Vďaka programu už by mal mať každý používateľ vždy najaktuálnejšiu verziu. Navyše vďaka on-line zobrazovaniu času priebehu príspevkov sa bude môcť pohodlne rozhodnúť, ktoré príspevky navštíviť.

Často sa stáva, že prezentujúci si neuvedomil, že čas na jeho príspevok uplynul. Pre vedúceho sekcie je často nepríjemný problém ho prerušiť, aby ho upozornil, že má končiť. Túto činnosť by mal zabezpečiť časovač, zobrazujúci zostávajúci čas do konca príspevku na vedľajšom monitore alebo tablete.

1.1.5 Charakteristika programu

Programu poskytne návštevníkom konferencie prehľadný rozvrh príspevkov pre všetky dni a miestnosti konferencie. Organizátorom bude poskytnutá možnosť tento rozvrh manažovať pred konferenciou a následne aj v reálnom čase počas konferencie. Vykonané zmeny sa návštevníkom premietnu do rozvrhu. Počas priebehu konferencie návštevníci v rozvrhuvidia, ktoré príspevky práve prebiehajú a koľko času zostáva do ich konca.

1.1.6 Identifikované roly

Používateľov programu rozdelíme do štyroch rolí. Prvou rolou je **návštevník konferencie**. Táto rola ako jediná nevyžaduje prihlásenie. Návštevník má právo len na prezeranie webstránky. Môže si prezerat' rozvrh, jednotlivé abstrakty a sťahovať články príspevkov. Nemôže nič meniť. Uvidí hotový rozvrh konferencie.

Druhou rolou je **organizačný výbor**. Organizačný výbor má najväčšie právomoci – manažuje rozvrh konferencie a prístupy ostatných rolí. Taktiež nahráva informácie o príspevkoch, ktoré sa budú na konferenciách prezentovať.

Ďalšou rolou je **vedúci sekcie**. Vedúci sekcie má právo manažovať rozvrh jemu pridelených sekcií. Jeho právomoci sú však menšie ako v prípade organizačného výboru. Môže zmeniť poradie príspevkov alebo zmeniť stav príspevku na neaktívny. Taktiež štartuje a ukončuje príspevky - informácia o práve prebiehajúcom príspevku sa premietne do rozvrhov návštevníkov konferencie.

Poslednou identifikovanou rolou je rola **používateľ časovača**. Po prihlásení si používateľ vyberie sekciu a následne sa na obrazovke zobrazí časovač tejto sekcie. Časovač na externom zariadení zobrazuje prezentujúcemu informáciu o zostávajúcom čase do konca príspevku.

1.2 Prehľad existujúcich systémov: MeetingHand

Požiadavky na program, ktorý vyvíjame v rámci tejto práce, sú detailné a unikátne. Nie je možné nájsť systém úplne odpovedajúci týmto požiadavkám. Hľadali sme preto systémy, ktoré sú podobné aspoň v niektorých častiach. Ako najpodobnejší systém našej aplikácii sme identifikovali MeetingHand.

MeetingHand je robustný systém umožňujúci manažovanie rôznych typov udalostí. Jedným z typov udalostí, ktorý je možné manažovať, je aj konferencia. Systém je na vyskúšanie 90 dní dostupný zdarma, ak by sme ho chceli využívať ďalej, je potrebné platiť poplatok. Najskôr sa treba zaregistrovať, následne máme možnosť vytvoriť ľubovoľné množstvo udalostí. Na každú udalosť môžeme zapísať 50 ľudí. Ak by sme chceli viac, je tiež potrebné platiť.

1.2.1 Vytvorenie udalosti

The screenshot shows the 'Event Main Settings' form in the MeetingHand application. The form is partially filled out, and a calendar widget is open over the date selection fields. The calendar shows the month of February 2020, with the 19th and 20th highlighted. The form includes the following fields and elements:

- Event Name***: A text input field with a red border and a red error message: "YOUR CHANGES COULD NOT BE SAVED. PLEASE COMPLETE THE EVENT LONG NAME SECTION TO CONTINUE." Below the field, it says "characters left: 80".
- Event Short Name***: A text input field with a red border and a red error message: "YOUR CHANGES COULD NOT BE SAVED. PLEASE COMPLETE THE EVENT SHORT NAME SECTION TO CONTINUE." Below the field, it says "characters left: 24".
- Event Start Date***: A date input field with the value "02.18.2020".
- Event End Date***: A date input field with the value "02.19.2020".
- Event Type***: A dropdown menu with the selected option "Free Event".
- Event Location & Venue S**: A text input field with the placeholder "Enter a location". Below it, there is a note: "Click above to enter the venue location manually".
- Calendar Widget**: A calendar for February 2020. The 19th is highlighted in green, and the 20th is highlighted in blue. The days of the week are labeled: SUN, MON, TUE, WED, THU, FRI, SAT.
- Navigation**: At the bottom left, there is a button "Cancel and Go Back" with a left arrow. At the bottom right, there is a button "NEXT" with a right arrow.

Obrázok č. 1: MeetingHand – Vytváranie udalosti

Pre vytvorenie udalosti musíme byť prihlásení ako organizátor. Ako vidíme na obrázku č. 1, je potrebné zadať názov udalosti, jej skrátený názov, dátum začiatku a konca, typ udalosti a nepovinne aj miesto. Dátum udalosti je zadávaný prostredníctvom prehľadného kalendára. Pri zadaní nesprávnych údajov sa formulár neodošle a dostaneme správy o chybách. Po úspešnom vytvorení udalosti sa dostaneme do prehľadu udalosti, ktorú sme práve vytvorili.

1.2.2 Prehľad organizátora o konferencií

Ako možno vidieť na obrázku č. 2, organizátorovi sa po prihlásení zobrazí obrazovka s veľkým množstvom informácií, zorientovanie sa chvíľu trvá. Nájdeme tu základné údaje o konferencií, počty žiadateľov o prijatie na konferenciu, počty už prijatých, dáta o registráciách, naše posledné aktivity a nastavenia. Aplikácia obsahuje veľké množstvo nastavení, takže si ju používateľ môže prispôsobiť svojim potrebám. Problém však nastane, ak má používateľ požiadavku, ktorá nie je zahrnutá v momentálnych možnostiach.

Dashboard [Settings](#)

CONFERENCE TIMELINE 1 DAYS LEFT

EVENT SETUP EVENT START DATE

Feb 17, 2020 Feb 19, 2020

CONFERENCE NUMBERS

	REGISTRATIONS	BOOKINGS
Total	0	0
Confirmed	0	0
Pending	0	0

REGISTRATIONS

REGISTRATION TYPE	FEE	PAID	PENDING	REMAINING QUOTA

PINNED REPORTS

You can pin reports from [Reports](#) page

ACTIVITY STREAM

Feb 17, 2020, Mon 13:28
J Mizerik (User)
 Mobile Phone custom form field name changed to

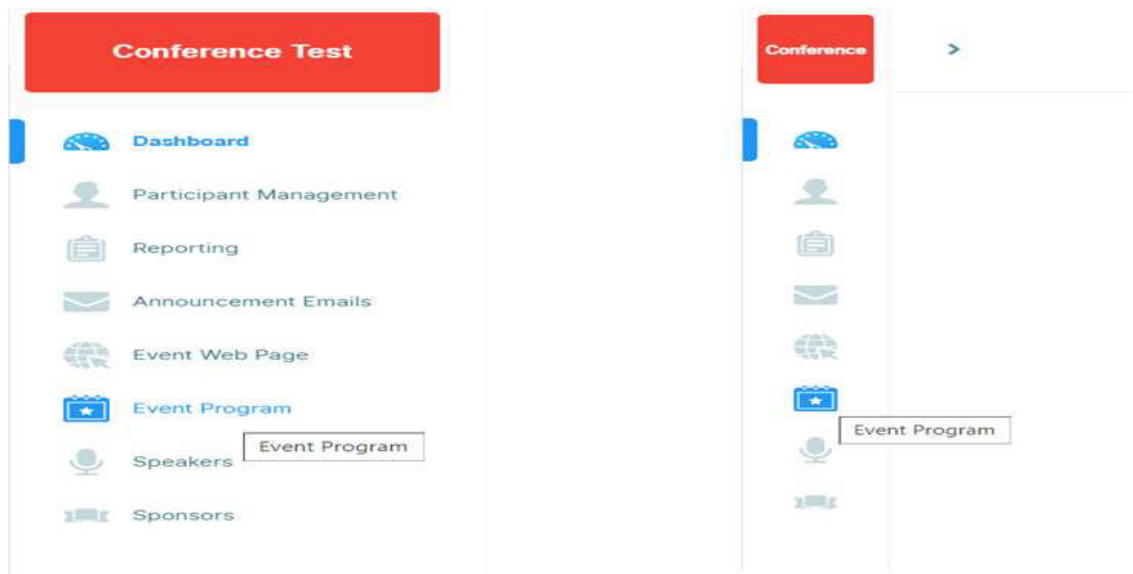
Feb 17, 2020, Mon 13:28
J Mizerik (User)
 Institution custom form field name changed to

Feb 17, 2020, Mon 13:28
J Mizerik (User)
 Title custom form field name changed to

Feb 17, 2020, Mon 13:28

Obrázok č. 2: MeetingHand – Prehľad prihláseného organizátora

Na obrázku č. 3 je bočná navigácia aplikácie. Kliknutím na čiernu šípku vľavo hore môžeme zmenšiť šírku navigácie. Vidíme potom len obrázky a názvy elementov sa objavia až po zameraní kurzorom na element. Navigácia je pri akejkoľvek veľkosti obrazovky vertikálna. U malej obrazovky by bola vhodnejšia horizontálna navigácia, prípadne možnosť ju úplne skryť.[2]

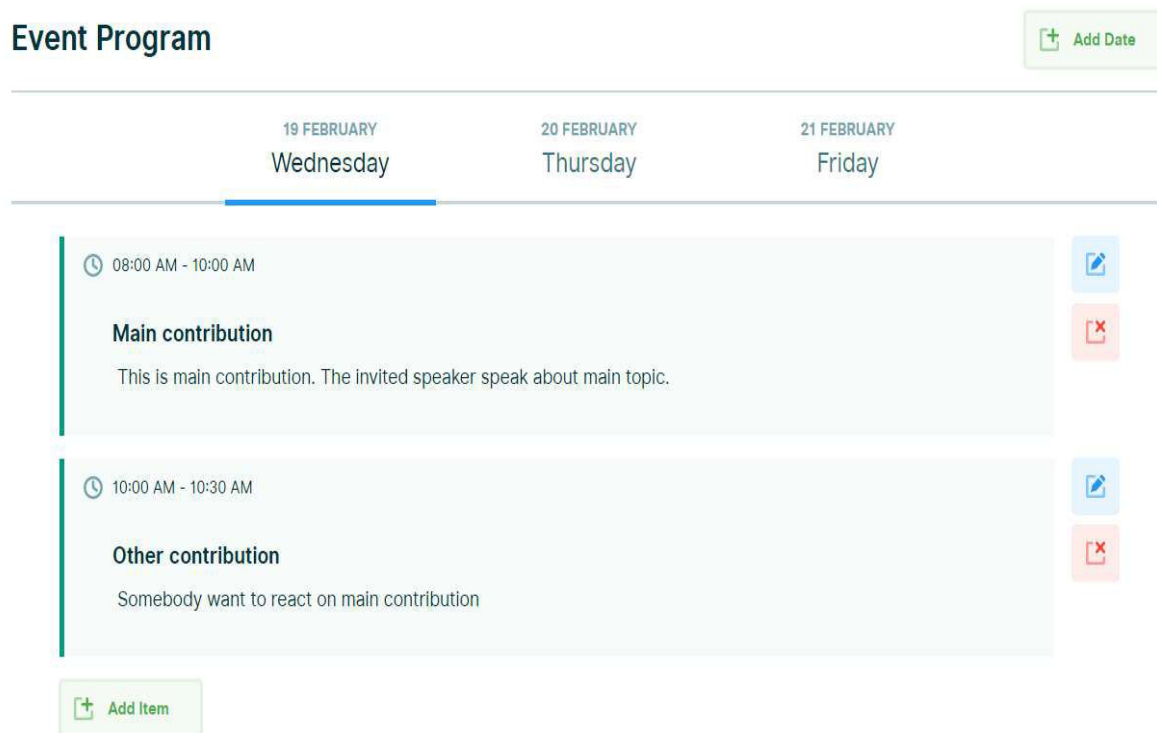


Obrázok č. 3: MeetingHand - Zmena bočného panela organizátora

V rámci hlavnej stránky konferencie má organizátor možnosť zobrazit' nasledujúce podstránky:

- Manažment členov – umožňuje manažovať prístupy účastníkov konferencie
- Reporty – možné stiahnuť súbory so štatistikami o konferenciách a účastníkoch
- Poslať e-mail – rozposlať e-mail účastníkovi alebo skupine účastníkov
- Webstránka udalosti – presmerovanie na webstránku udalosti z pohľadu účastníka
- Program udalosti – manažovanie rozvrhu konferencie
- Rečníci – manažovanie účtov pre rečníkov
- Sponzori – možnosť pridať sponzora a jeho logo zobrazit' na webstránke

1.2.3 Vytváranie rozvrhu



Obrázok č. 4: MeetingHand - Vzhľad rozvrhu z pohľadu organizátora

Na obrázku č. 4 vidíme tvorbu rozvrhu. Medzi dňami možno prepínať v hornej lište, samotné položky rozvrhu sa zobrazujú len pod sebou. Chýba podpora pre viac miestností. Ak sa konferencia uskutočňuje vo viacerých miestnostiach, je nutné vytvoriť novú udalosť. Veľkosť jednotlivých buniek sa neprispôsobuje dobe trvania príspevku.

Obrázok č. 5 znázorňuje pridávanie prvku rozvrhu. Čas vyberáme z predpripraveného formulára, kde je nutné „scrollovať“, aby sme sa dostali k nami požadovanej hodnote. Dostať sa napríklad k hodnote 16:30 trvá dlho. Rečník príspevku je pridávaný kliknutím na tlačidlo a otvorením ďalšieho formulára.

Add Agenda Item [X]

Start Time: 10:00 AM

End Time: 10:00 AM

Date: 2020-02-19

Title: Other contribution

Item Description: Somebody want to react on main contribution

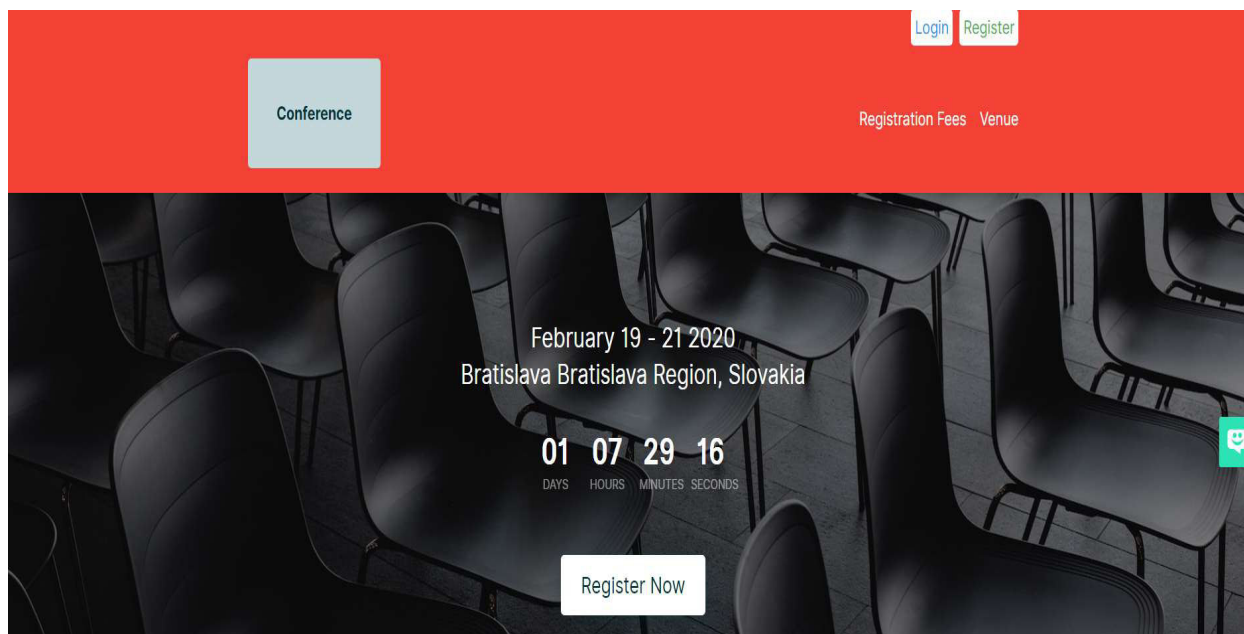
Speakers: Pick speaker [v] [Add Speaker]

Location: _____

Obrázok č. 5: MeetingHand - Vytváranie položky rozvrhu a možnosť voľby času vo formulári

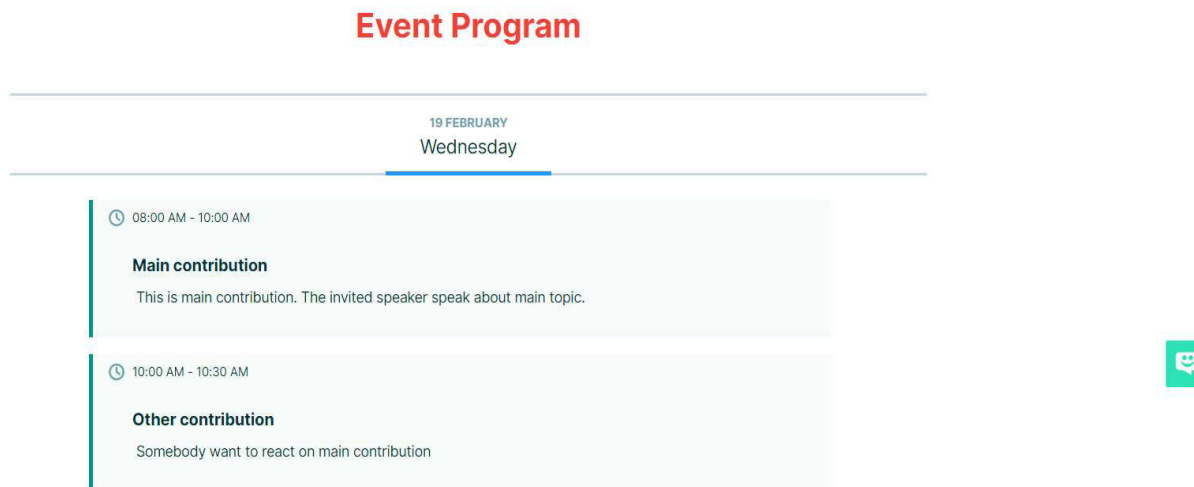
1.2.4 Webstránka z pohľadu návštevníka

Na obrázku č. 6 je znázornená webstránka z pohľadu návštevníka. Vidíme základné údaje o konferencii – jej názov, dátum, miesto. V prípade, že konferencia ešte nezačala, vidíme aj odpočítavanie času do jej začiatku. Ponúka sa možnosť novej registrácie návštevníka.



Obrázok č. 6: MeetingHand - Stránka z pohľadu návštevníka

Na stránke nájdeme aj rozvrh pre návštevníka konferencie. Design je veľmi podobný ako v prípade rozvrhu pre organizátora konferencie. Návštevník však nemôže vykonávať úpravy.



Obrázok č. 7: MeetingHand - Rozvrh z pohľadu návštevníka

Ako nedostatok možno vnímať skutočnosť, že celá aplikácia je založená na „vyskakujúcich“ formulároch. Často sa stáva, že musíme otvoriť až štyri formuláre, kým sa dostaneme k požadovanej funkcionalite. Pri používaní aplikácie prakticky celý čas klikáme na tlačidlá, ktoré otvárajú nové okná. Ďalším nedostatkom systému je, že okrem zmenšenia navigácie, nie je vôbec prispôsobený pre zariadenia s menšou obrazovkou. Elementy sa neprispôsobujú a tak je pri používaní tabletu či mobilu nutné často „scrollovať“. U takéhoto typu aplikácie, ktorý potenciálne môže obsahovať veľké množstvo dát, je vytvorenie dynamicky reagujúceho obsahu náročnejšie, no aspoň nejaká podpora by mohla byť dostupná.[3]

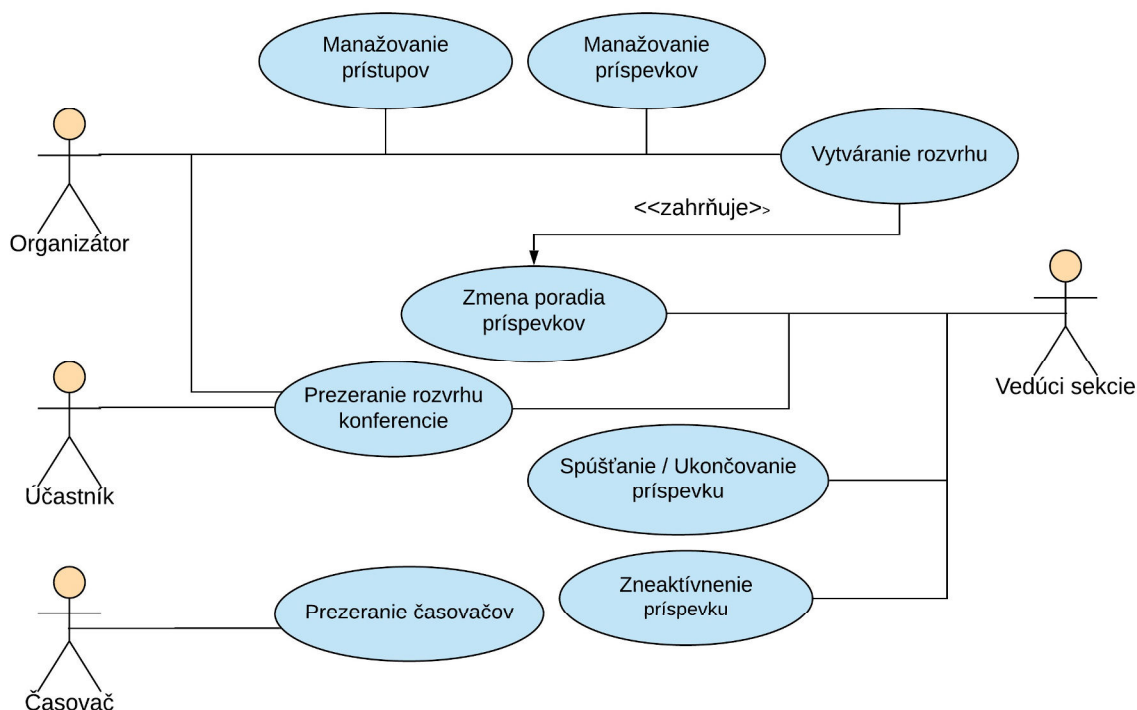
V tejto kapitole sme si bližšie predstavili priebeh konferencie. Pochopenie priebehu nám pomôže v ďalších častiach práce pri návrhu a následnej implementácii systému. Špecifikovali sme si základné požiadavky zadávateľa na systém, ktoré si bližšie rozanalyzujeme v nasledujúcej kapitole. Taktiež sme si analyzovali systém MeetingHand. V ďalšej kapitole porovnáme požiadavky nášho programu s týmto systémom. Následne vyberieme riešenia, ktoré vyhovujú aj nášmu systému a podľa potreby ich využijeme pri implementácii.

2 Analýza požiadaviek a návrh systému

V rámci špecifikácie požiadaviek sme používateľov programu rozdelili do štyroch rolí. V tejto kapitole si jednotlivé roly roanalyzujeme podrobnejšie. Následne vytvoríme návrh systému, ktorý bude vychádzať z vykonanej analýzy, štúdia zdrojov a praktických skúseností autora.

2.1 Charakteristika rolí

Základné rozdelenie právomocí jednotlivých rolí, vychádzajúce zo špecifikácie, vidíme na obrázku č. 8. V nasledujúcich podkapitolách si roly popíšeme podrobnejšie.



Obrázok č. 8: Use Case diagram

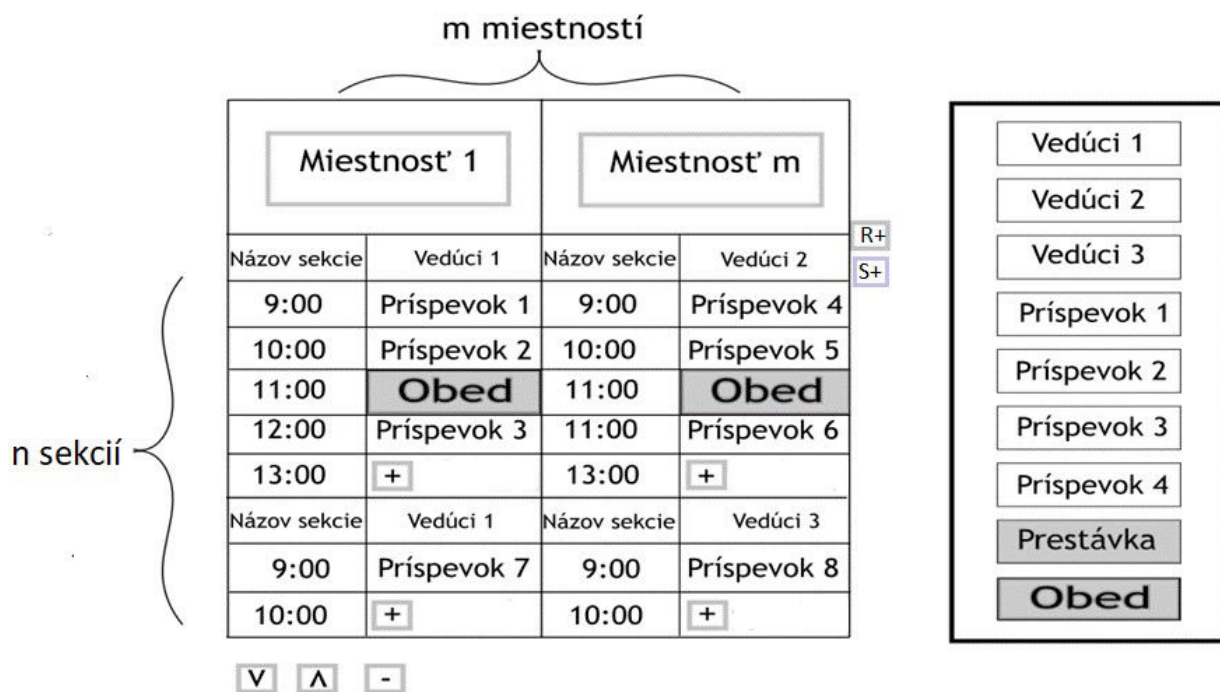
2.2 Rola organizačný výbor

Organizačný výbor má po prihlásení prístup na šesť podstránok. Každú z nich si popíšeme.

2.2.1 Úprava rozvrhu

Najskôr je potrebné novú konferenciu založiť a vložiť názov konferencie a jej dátum. Vložený údaj o názve konferencie je následne uvedený v hlavičke webstránky. Až po vložení týchto údajov sa prihlásený člen organizačného výboru dostáva k tvorbe rozvrhu. Prepínanie medzi jednotlivými dňami konferencie zrealizujeme rovnako ako náš analyzovaný systém MeetingHand - prostredníctvom záložky.

Pre vytvorenie rozvrhu je potrebné zadať do systému osoby, ktoré sa konferencie zúčastnia. To sa vykoná na podstránke Členovia, popísanej v 2.2.2. Na základe zadaných osôb je pri tvorbe rozvrhu k dispozícii bočný panel. Bočný panel obsahuje malé elementy – „kartičky“ vedúcich sekcií a samotných príspevkov. Na konci radu príspevkov sú ešte špeciálne kartičky odlíšené inou farbou – prestávka na kávu, obed, večeru. Z daného zoznamu je následne možné vedúceho sekcie alebo príspevok potiahnutím umiestniť priamo do príslušnej bunky rozvrhu.



Obrázok č. 9: Náskres tvorby rozvrhu konferencie

Na obrázku č. 9 vidíme upravovanie rozvrhu pre rolu organizátor. Každý prvok rozvrhu sa skladá z dvoch častí – prvá časť uvádza čas, kedy príspevok začína, druhá časť je najskôr

prázdna - neskôr tam budeme pomocou myši presúvať príspevok z bočného panelu. Časy začiatkov príspevkov sa zaokrúhľujú na päť minút. Vkladanie nekorektných časov je potrebné ošetriť. Kliknutím na tlačidlo „+“ umiestnené pod posledným riadkom rozvrhu pridávame ďalšie prvky rozvrhu.

Je potrebné, aby organizátor mohol vložiť príspevok kdekoľvek do rozvrhu bez toho, aby sme museli presúvať všetky, už dovedy vložené príspevky. Funkcionalita bude zabezpečená pomocou dvoch šípok dole/hore. Každý prvok rozvrhu je možné kliknutím označiť. Po kliknutí na prvok je používateľ schopný pomocou týchto tlačidiel prvok presúvať. Presúva sa len časť prvku rozvrhu s príspevkom. Časy sa automaticky prepočítajú. Pomocou tlačidla „-“ zmažeme označený prvok rozvrhu.

Rozvrhy jednotlivých miestností by sa mali zobrazovať rovnobežne vedľa seba podľa rovnakých časov, čo sprehl'adni rozvrh používateľovi. Problémom pri takomto zobrazení je, že časovanie príspevkov jednotlivých miestností môže byť rôzne. Teda napríklad v jednej miestnosti môže byť príspevok medzi 9:00-10:00, ale v inej dva príspevky od 9:00 do 9:30 a od 9:30 do 10:00. Preto veľkosť jednotlivých buniek rozvrhu bude závisieť od času príspevku. Ako vidíme na obrázku č. 10, príspevky v prvej miestnosti trvajú 1 hodinu, príspevky v druhej miestnosti až 1 hodinu a 30 minút. Výška prvkov rozvrhu v druhom prípade je preto vyššia vzhľadom na požiadavku rovnobežného zarovnania.

Miestnosť 1		Miestnosť 2	
Názov sekcie	Vedúci 1	Názov sekcie	Vedúci 2
9:00	Príspevok 1	9:00	Príspevok 4
10:00	Príspevok 2	10:30	Príspevok 5
11:00	Príspevok 3		
13:00	+	13:00	+

⬇
⬆
-

Vedúci 1

Vedúci 2

Vedúci 3

Príspevok 1

Príspevok 2

Príspevok 3

Príspevok 4

Prestávka

Obed

Obrázok č. 10: Zarovnanie rôzne dlhých príspevkov v rovnaký deň

Potrebuje nájst' spôsob, ktorým by bolo možné rozdeliť miestnosti na viacero sekcií. Zároveň požadujeme, aby rovnobežné zobrazenie príspevkov v rovnakom čase zostalo zachované. Jednou z alternatív, ktoré sme analyzovali, bolo vkladanie novej sekcie pomocou elementu z bočného panelu – rovnako, ako sa do rozvrhu vkladajú príspevky a vedúci sekcie. Tento bočný panel však už v terajšej situácii obsahuje vedúcich sekcie, prestávky a samotné príspevky. Pridanie ďalšieho typu elementu by ho spravilo príliš obsiahlym a neprehľadným. Navyše takéto vloženie názvu sekcie do rozvrhu by prinášalo problém, kde tento názov umiestniť. Na všetkých miestach tento názov pôsobí neprehľadne. Ďalšou alternatívou by bolo vloženie polí s názvami sekcií na začiatok dňa a následne jednotlivé sekcie farebne odlišiť, čím by sme ich identifikovali. Pri väčšom počte sekcií by však rozvrh pôsobil príliš farebne. Navyše nie je príliš praktické, keď je názov sekcie ďaleko od samotnej sekcie. Napokon sme rozhodli, že nové sekcie sa budú pridávať pomocou šablóny. Keďže sa často stáva, že dĺžky príspevkov jednotlivých sekcií sú podobné, šablóna skopíruje prvky rozvrhu už vytvorených sekcií a vytvorí rovnakú skupinu. V takom prípade nám stačí vytvoriť jednu skupinu sekcií a následne pomocou šablóny túto skupinu sekcií skopírujeme. Ak sa nová skupina sekcií časovo nezhoduje so skupinou sekcií, ktorú sme kopírovali, jednoducho dorobíme alebo zmažeme niekoľko prvkov rozvrhu, prípadne upravíme ich čas.

Ďalšou požiadavkou je, aby sa čo najviac miestností zobrazovalo vedľa seba, čo používateľovi sprehľadní situáciu. Ak je počet miestností príliš veľký na zobrazenie na jednej stránke po šírke, bude sa radšej „scrollovať“ akoby sa vytváral nový riadok. Nový riadok sa bude vytvárať iba v prípade novej skupiny sekcií. Výsledný rozvrh bude teda zobrazený po riadkoch vzhľadom na sekcie a po stĺpcoch vzhľadom na miestnosti.

Polia v rozvrhu, ktoré je nutné vyplniť a ešte nie sú vyplnené, budú zvýraznené špeciálnou farbou.

2.2.2 Členovia

Táto podstránka slúži k manažovaniu prístupov pre členov organizačného výboru a pre vedúcich sekcie. Je tu možnosť pridávať nových ľudí, ale v prípade potreby aj upravovať údaje už existujúcich členov alebo ich úplne zmazať.

Prvým typom účtu je člen organizačného výboru – na začiatku existuje iba jediný člen organizačného výboru, označený ako hlavný admin. Pri prvom spustení programu je prednastavené jeho prihlasovacie meno „admin” a heslo rovnako „admin”. Prihlasovacie meno a heslo je po prihlásení možné zmeniť v sekcii nastavenia. Hlavný admin môže vytvoriť ľubovoľné množstvo ďalších členov organizačného výboru. Ak vytvárame člena organizačného výboru, je potrebné zadať jeho celé meno, prihlasovacie meno a heslo. Dĺžka hesla musí byť minimálne 5 znakov. Spravovať účty organizátorov môže iba hlavný admin. Všetci ostatní organizátori môžu spravovať účty vedúcich sekcie, spravovať príspevky, upravovať rozvrh a nastavenia konferencie.

Ďalším typom účtu je vedúci sekcie, tzv. „chair“. Jeho právomoci sú konkrétnejšie popísané v podkapitole 2.3. Pre vytvorenie vedúceho sekcie treba zadať jeho celé meno. Nepovinne možno zadať aj prihlasovací kód. Keďže vedúceho sekcie nechceme zaťažovať rôznou administratívou, nevytvárame mu heslo. Do systému sa prihlási pomocou prihlasovacieho kódu, ktorý mu odovzdá organizačný výbor. Prihlasovací kód je ľubovoľný reťazec s dĺžkou minimálne 5. Ak pri vytváraní prístupu pre vedúceho sekcie prihlasovací kód nie je zadaný, program ho vygeneruje automaticky ako náhodné unikátne desaťmiestne číslo.

2.2.3 Príspevky

Táto podstránka slúži na vkladanie príspevkov konferencie. Príspevky pridáme vyplnením jednoduchého formulára, kde uvedieme názov príspevku, autora alebo autorov a nepovinne aj abstrakt príspevku. Taktiež môžeme nahrať PDF súbor s článkom. Vytvorené príspevky možno upravovať a mazať.

2.2.4 Nastavenia

V tejto sekcii možno zmeniť základné údaje o konferenciách, upraviť nastavenia rozvrhu a časovača. Zmeniť možno názov konferencie, dátum konania konferencie, veľkosť buniek rozvrhu, predvolenú dĺžku prvého príspevku v sekcii, predvolenú dĺžku ostatných príspevkov a heslo k časovaču.

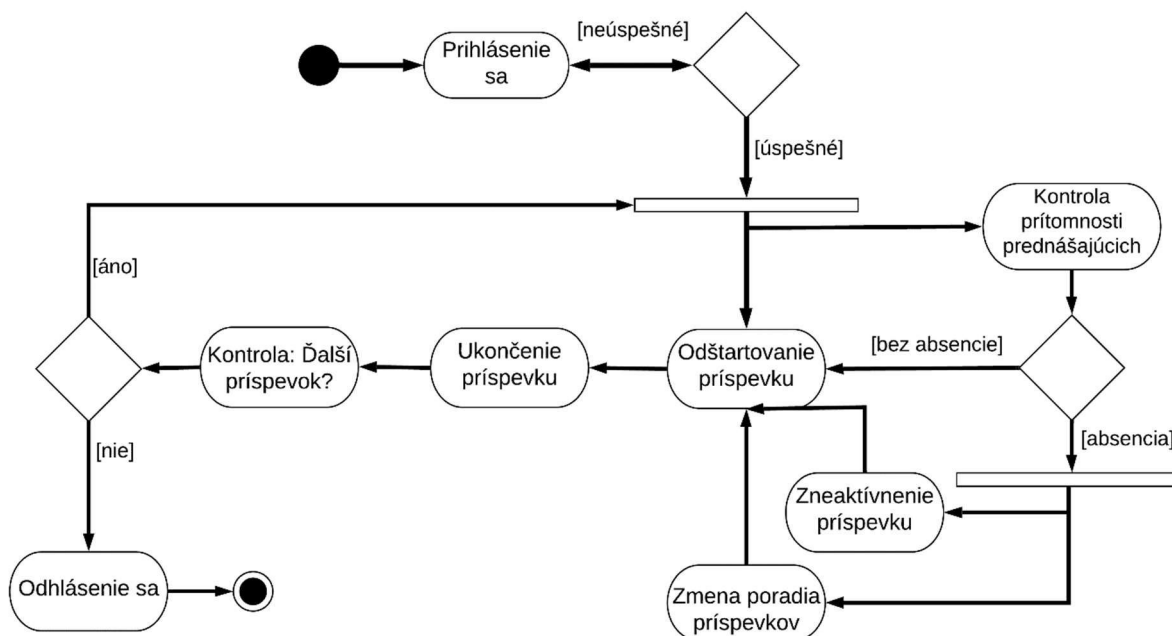
2.2.5 Rozvrh

Na tejto podstránke sa organizátorovi zobrazí rovnaký rozvrh, ako uvidia návštevníci konferencie. Cieľom je, aby sa nemusel odhlasovať, pokiaľ chce vidieť finálny stav rozvrhu.

2.2.6 Nastavenia účtu

Táto podstránka umožní prihlásenému členovi organizačného výboru zmeniť si údaje svojho účtu. Môže si zmeniť heslo k svojmu účtu, svoje prihlasovacie meno a svoje skutočné meno, ktoré sa zobrazuje v aplikácii. Taktiež má možnosť odhlásiť sa zo všetkých zariadení.

2.3 Rola vedúceho sekcie



Obrázok č. 11: Diagram aktivít – aktivity vedúceho sekcie počas konferencie

Činnosti vedúceho sekcie počas konferencie zobrazuje diagram aktivít na obrázku č. 11. [4] Po prihlásení sa vedúcemu zobrazia všetky sekcie, ktoré má na starosti. Ich názvy spoločne s dátumom a miestnosťou. Po kliknutí na jednu zo sekcií sa vedúci dostane k rozvrhu. Vzhľad rozvrhu vedúceho sekcie možno vidieť na obrázku č. 12. Vedúci sekcie najskôr kliknutím

na tlačidlo „start“ príspevok odštartuje. Tým aktivuje daný príspevok, ktorý bude v rozvrhu pre návštevníkov zvýraznený ako práve prebiehajúci. Po začatí príspevku sa tlačidlo „start“ zmení na tlačidlo „stop“. Počas príspevku vedúci sekcie a prezentujúci vidia časovač, ktorý znázorňuje zostávajúci čas príspevku. Čas je najskôr znázornený zelenou farbou, posledných 5 minút sa farba zmení na bielu. Biela farba označuje, že príspevok by sa už mal skončiť a prebiehať by mala diskusia. V momente, keď čas vyprší by diskusia už mala byť ukončená. Následne vedúci sekcie klikne na tlačidlo „stop“, čím sa informácia o ukončení premietne do rozvrhu všetkým návštevníkom.

Miestnosť x	
Názov sekcie 1	Vedúci 1
8:00	Príspevok 1
9:00	Príspevok 2 (36:12)
10:00	Príspevok 3
11:00	Príspevok 4
12:00	

Obrázok č. 12: On-line manažovanie – vedúci sekcie

Vedúci sekcie môže upraviť rozvrh svojich sekcií. Upravovanie rozvrhu pre vedúceho sekcie má rovnaký formát ako upravovanie rozvrhu pre organizačný výbor, jeho právomoci sú však obmedzené. Môže meniť poradie príspevkov alebo označiť príspevky za neaktívne. Pozadie neaktívneho príspevku sa stane tmavším a objaví sa informácia o zrušení.

2.4 Rola používateľ časovača

Rola používateľ časovača sa využíva pre prístup k časovačom jednotlivých sekcií. Zvyčajne túto činnosť vykonáva vedúci sekcie, výnimočne to však môže byť aj iná poverená osoba. Po prihlásení do tejto roly sa zobrazia všetky sekcie priebehu konferencie v daný deň. Po zvolení konkrétnej sekcie sa na celú obrazovku zobrazí časovač. Tento časovač sa zobrazuje

na vedľajšom monitore alebo tablete tak, aby ho dobre videl prednášajúci. Jeho úlohou je zobrazovať prednášajúcemu čas, ktorý zostáva do konca jeho príspevku.

2.5 Rola návštevníka

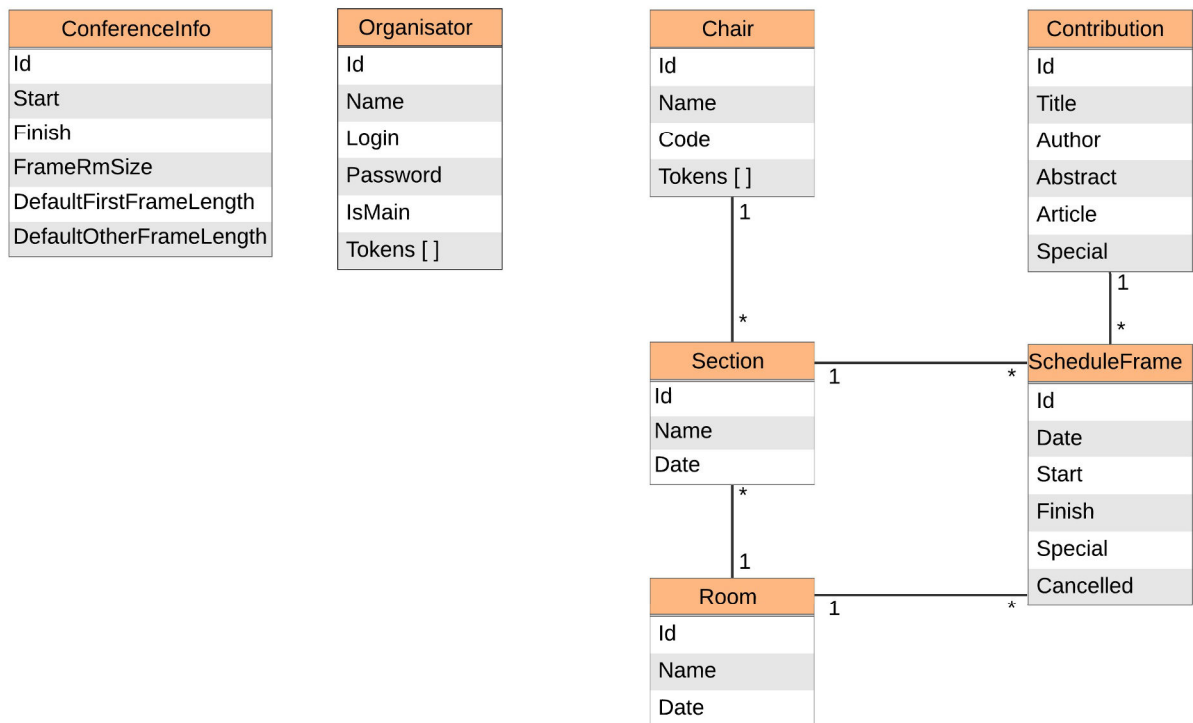
Ide o používateľa, ktorý má prístup k aktuálnej verzii webovej stránky konferencie. Pre rolu návštevníka nie je potrebná autentifikácia. Predpokladá sa, že stránku navštívia primárne účastníci konferencie, no teoreticky ju môže navštíviť ktokoľvek na internete. Návštevník si stránku môže ľubovoľne prezerat', nemá však možnosť meniť jej obsah.

Na stránke je už pred konferenciou dostupný predpokladaný rozvrh priebehu konferencie. V rámci rozvrhu je možné vidieť jednotlivé dni konferencie, miestnosti, zaradenie príspevkov do sekcií a vedúcich jednotlivých sekcií, časy jednotlivých príspevkov, názvy príspevkov, ich autorov. Príspevky, ktoré sú v rovnakom čase, by mali byť zarovnané do rovnakého riadku, čo by malo zvýšiť prehľadnosť pre návštevníkov konferencie. Po kliknutí na názov príspevku, resp. ikonu abstraktu sa zobrazí abstrakt príspevku. Pokiaľ má príspevok nahratý článok, návštevník si ho môže stiahnuť.

Následne v čase konferencie je dostupný rovnaký rozvrh ako pred konferenciou, no navyše sa v reálnom čase v rozvrhu zobrazuje, ktorý príspevok práve prebieha a koľko času zostáva do jeho konca.

2.6 Modely

Keď už máme analyzované jednotlivé roly, môžeme vytvoriť modely tried a charakterizovať vzťahy medzi nimi. Diagram tried modelov vidíme na obrázku č. 13.



Obrázok č. 13: Diagram tried

Medzi modelmi existuje päť relácií. Model ScheduleFrame reprezentuje jeden prvok rozvrhu. Jeho atribútmi sú dátum, čas začiatku, čas konca, informácia, či sa jedná o špeciálnu bunku - akou je napríklad prestávka, a informácia o tom, či bunka rozvrhu bola zrušená. Model Contribution reprezentuje príspevok vložený do systému a jeho atribúty obsahujú informácie o tomto príspevku. Atribúty Abstract a Article sú nepovinné, takže môžu mať aj hodnotu null. Abstrakt ukladá samotný textový reťazec abstraktu, Article ukladá adresu článku, ktorý máme uložený na servery. Pokiaľ nedôjde k vloženiu príspevku do samotného rozvrhu, vzťah medzi Contribution a ScheduleFrame nemusí existovať. Veľmi podobný je vzťah medzi modelmi Chair a Section. Chair reprezentuje vedúceho sekcie. Okrem atribútu mena a prihlasovacieho kódu, obsahuje aj pole Tokens, v ktorom sa ukladajú prihlasovacie tokeny, ktoré budú bližšie popísané v podkapitole 2.9.2 Autentifikácia a autorizácia. Model Section je vo vzťahu s ľubovoľne veľa inštanciami modelu ScheduleFrame pretože sekcia môže obsahovať ľubovoľne veľa prvkov rozvrhu, ktoré zase zvyčajne obsahujú príspevky. Model Room je vo vzťahu s ľubovoľne veľa inštanciami modelu Section, pretože miestnosť môže obsahovať ľubovoľne veľa sekcií.

Model ConferenceInfo je singleton – má jediná inštanciu, v ktorej sú uložené základné údaje a nastavenia konferencie. Táto entita nie je vo vzťahu so žiadnym ďalším modelom. V atribútoch modelu organizátor ukladáme meno, prihlasovacie údaje organizátora, informáciu o tom, či sa jedná o hlavného admina a tiež pole prihlasovacích tokenov. Model organizátor nie je v relácií so žiadnym iným modelom.

2.7 Použité technológie a nástroje

V tejto podkapitole si predstavíme technológie a nástroje použité pri vývoji aplikácie.

2.7.1 HTML, CSS, JavaScript

Na front end aplikácie používame HTML5, CSS3 a JavaScript vo verzii ECMAScript 6.

2.7.2 Node.js

Node.js umožňuje vytvoriť webový server použitím programovacieho jazyka JavaScript. Skutočnosť, že stranu servera aj stranu klienta máme v rovnakom programovacom jazyku je benefitom. Pomocou Node.js môžeme komunikovať s prehliadačom a vytvoriť dynamický obsah ešte predtým, ako sa webstránka prehliadaču odošle.

JSON formát je najčastejšie využívaný formát pre transport dát na internete a JavaScript má na prenos tohto formátu zabudovanú podporu. Použitie JavaScriptu teda znamená, že transport dát v JSON formáte je rýchly a jednoduchý.[5]

Špecifické pre Node.js je, že využíva jednovláknové programovanie. Jediné vlákno zaberie menej pamäte v porovnaní s prípadom, keď by vlákien bolo viac. Keďže sa všetok kód vykonáva sekvenčne v jedinom vlákne, akcie, ktoré trvajú dlho, treba vykonávať asynchrónne. To je možné docieľiť pomocou „callbackov“, čo však môže viesť k problému tzv. „callback hell“. Novším a preferovanejším prístupom je využitie „promises“.

Node.js pre zjednodušenie tvorby aplikácií využíva systém modulov. Požadovaný modul možno pridať do svojej aplikácie a tým využiť jeho funkcionality. Node.js obsahuje len niekoľko základných modulov, ostatné je potrebné do aplikácie doinštalovať.[6]

2.7.3 NPM – Node manažér balíčkov

NPM je správca knižníc pre Node.js. V tejto podkapitole popíšeme knižnice, ktoré budú využité v aplikáciách.

Express

Express je malá knižnica zjednodušujúca prácu s Node.js serverom. Zjednodušuje organizáciu kódu, pridáva pomocnú funkcionality Node.js HTTP objektom.[7]

Socket.IO

Socket.IO je knižnica poskytujúca rozhranie na prepojenie servera s klientom. Toto prepojenie je obojsmerné a realizované posielaním správ medzi Node.js serverom a prehliadačom klienta. Jedna strana čaká na určitú udalosť. Keď druhá strana vyšle správu o tejto udalosti spoločne s potrebnými dátami, prvá strana môže zareagovať.[8]

Bcryptjs

Knižnica poskytuje funkcionality na zakódovanie hesla. Používa Bcrypt algoritmus, ktorý umožňuje jednosmerné zakódovanie. Po zakódovaní už neexistuje algoritmus, ktorým by sme zo zakódovaného reťazca získali pôvodný. Pri kontrole správnosti hesla sa zadané heslo opäť zakóduje a obe zakódované heslá sa porovnávajú.[9]

JSONWebtoken

Táto knižnica slúži na autentifikáciu a autorizáciu používateľov. Dokáže vytvárať znaky, tzv. tokeny, ktorými používateľ potvrdzuje svoju identitu. K vytvoreniu tokenu dochádza po úspešnom prihlásení používateľa, po odhlásení zase k jeho zmazaniu. Je potrebné, aby tokeny boli unikátne, čo je garantované pomocou algoritmu HS256.[10]

Cookie-parser

Poskytuje nástroje na prácu s „cookies“. Dokáže prečítať hlavičky transakcií medzi klientom a serverom. Následne cookie označí príslušným názvom a uloží.

Ejs

Pomocou tejto knižnice možno vytvárať HTML webstránky s dynamickým obsahom. Poskytuje šablónu, do ktorej môžeme vkladať javascriptový kód. Koncovku takéhoto súboru treba upraviť na .ejs.[11]

Timepicker

Je malá knižnica, poskytujúca možnosť praktického vkladania času do formulára. Nastavenia si je možné prispôsobiť svojim potrebám.

Mongoose

Mongoose je knižnica na vytváranie modelov a prácu s nimi. Je špeciálne navrhnutá pre prácu v asynchrónnom prostredí. Podporuje asynchrónne programovanie pomocou „callbacks“ aj „promises“. Obsahuje zabudovanú podporu pre konverziu typov, validáciu, vytváranie dopytov. [12].

Mongodb

Knižnica poskytujúca rozhranie pre prepojenie Node.js a Mongodb databázy. Mongodb databáza je bližšie popísaná v nasledujúcej podkapitole.

2.7.4 MongoDB

MongoDB je nástroj na permanentné ukladanie dát. Permanentné znamená, že dáta ostávajú zachované aj po vypnutí programu. Ide o tzv. NOSQL databázu, nie je využívaný jazyk SQL. Keďže v našej aplikácii máme medzi entitami len niekoľko málo relácií, vybraná bola práve databáza Mongo.db. Node.js prostredníctvom svojich knižníc poskytuje vysokú kompatibilitu s touto databázou.

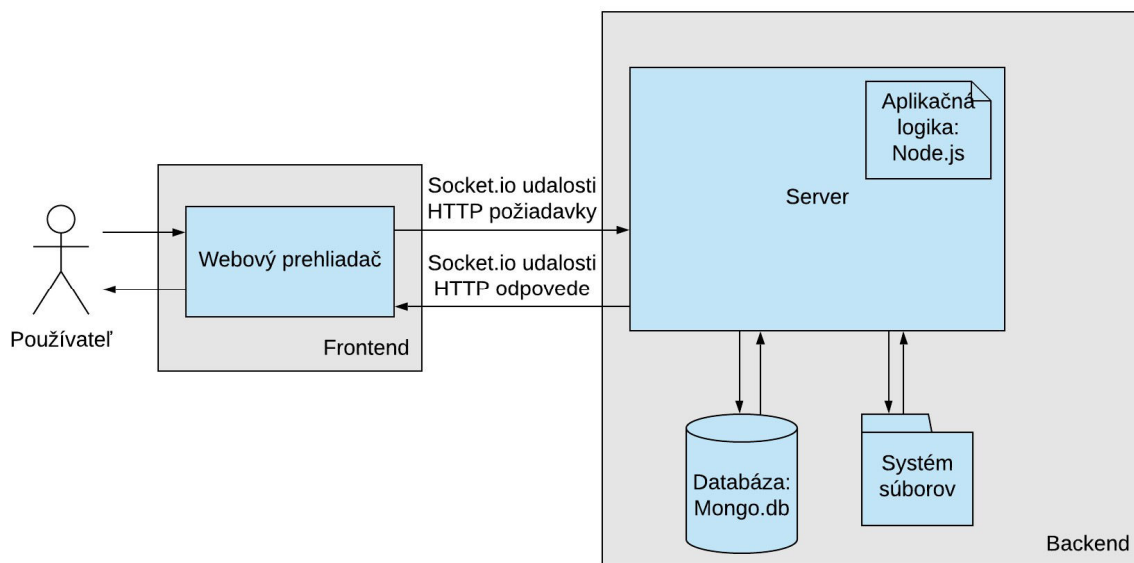
Výhodou výberu tejto databázy je skutočnosť, že do databázy môžeme ukladať priamo javascriptové objekty. S využitím knižnice Mongoose ukladáme priamo objekty jednotlivých

modelov, nepotrebujeme žiadne mapovanie na databázu. Každú kolekciu v databáze reprezentuje jeden model. Ďalšou výhodou je, že vďaka niekoľkým Node.js knižniciam dokážeme veľmi jednoducho vytvoriť autorizačný systém, ktorý je popísaný v podkapitole 2.9.2. Táto databáza podporuje ukladanie do polí, čo tiež využijeme. [13]

Napriek skutočnosti, že ide o NOSQL databázu, databáza poskytuje koncept primárnych a cudzích kľúčov a potrebu spájania tabuliek možno nahradiť agregáciou kolekcí. Agregácia by mohla byť pomalšia pri veľmi veľkých množstvách dát, vybraných z veľkého množstva kolekcí. V našej aplikácii však takáto situácia nenastáva.[14]

2.8 Architektúra systému

Architektúru systému zobrazuje obrázok č. 14. Aplikácia je rozdelená na frontend a backend. Používateľ s aplikáciou komunikuje prostredníctvom webového prehliadača.

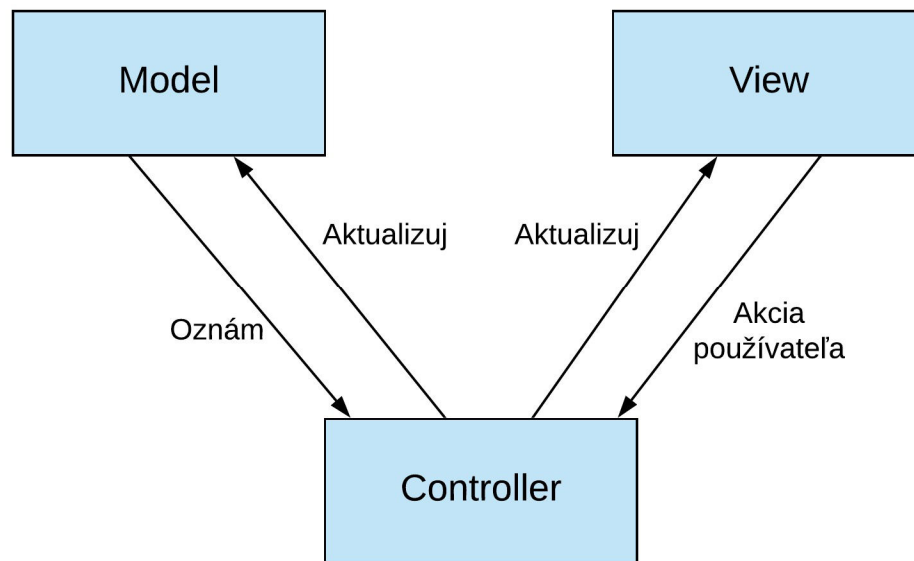


Obrázok č. 14: Architektúra systému [15]

Ide o systém založený na udalostiach. Na princípe udalostí funguje nie len GUI aplikácie, ale aj komunikácia medzi webovým prehliadačom (klientom) a serverom. Táto komunikácia prebieha cez Socket.io. Hlavným prínosom tejto technológie je skutočnosť, že vykonané zmeny dokážeme klientovi zobrazit' bez potreby aktualizácie stránky.

Príslušné funkcie na servery čakajú na istú pomenovanú udalosť. Dokážu zareagovať v momente, ako od klienta dostanú správu, že k danej udalosti došlo. V závislosti od výsledku operácie môže server poslať správu späť klientovi, ktorý tiež na takúto udalosť čaká. Podľa konkrétnej situácie, server posiela správu buď len klientovi, ktorý udalosť vyvolal, alebo všetkým klientom okrem klienta, ktorý udalosť vyvolal, prípadne úplne všetkým klientom.

Jedinou výnimkou, keď na komunikáciu medzi klientom a serverom nepoužívame Socket.io, sú routes. Tie fungujú na princípe http požiadaviek a odpovedí. Routes sú bližšie popísané v podkapitole 2.9.2 Autentifikácia a autorizácia.



Obrázok č. 15: Použitie návrhového vzoru MVC v aplikácii

Pri návrhu systému sme postupovali podľa návrhového vzoru MVC, ktorý aplikáciu rozdeľuje na tri hlavné časti – model, view, controller. Využitím tohto návrhového vzoru sme dosiahli čistejší kód, vyššiu prehľadnosť a vyššiu modularitu systému.[16]

Aplikácia si stav modelu udržiava v databáze a čiastočne aj na servery. Na servery máme uložené PDF súbory článkov a v čase priebehu konferencie aj časovač a dáta o čase štartu príspevkov. Controller možno považovať za sprostredkovateľa interakcie medzi dvoma zvyšnými časťami. Controller s databázou komunikuje prostredníctvom servera. View,

na základe pokynov controllera, udržiava používateľské rozhranie a informuje používateľa o zmenách stavu modelu.[17]

2.9 Server

Aplikácia funguje na servere Node.js, ktorého výhody boli popísané v podkapitole 2.7.2.

2.9.1 Komunikácia s databázou

Komunikácia s databázou trvá určitý, vopred neznámy čas. Dĺžka tohto času závisí od toho, akú operáciu vykonávame. JavaScript neposkytuje viacvláknové programovanie, my však potrebujeme, aby program pokračoval vo svojom behu aj v dobe, keď čaká na odpoveď od databázy. Tento problém vyriešime pomocou asynchrónnych funkcií, ktoré umožnia, aby zvyšok programu bežal, zatiaľ čo čakáme na odpoveď od databázy. [18]

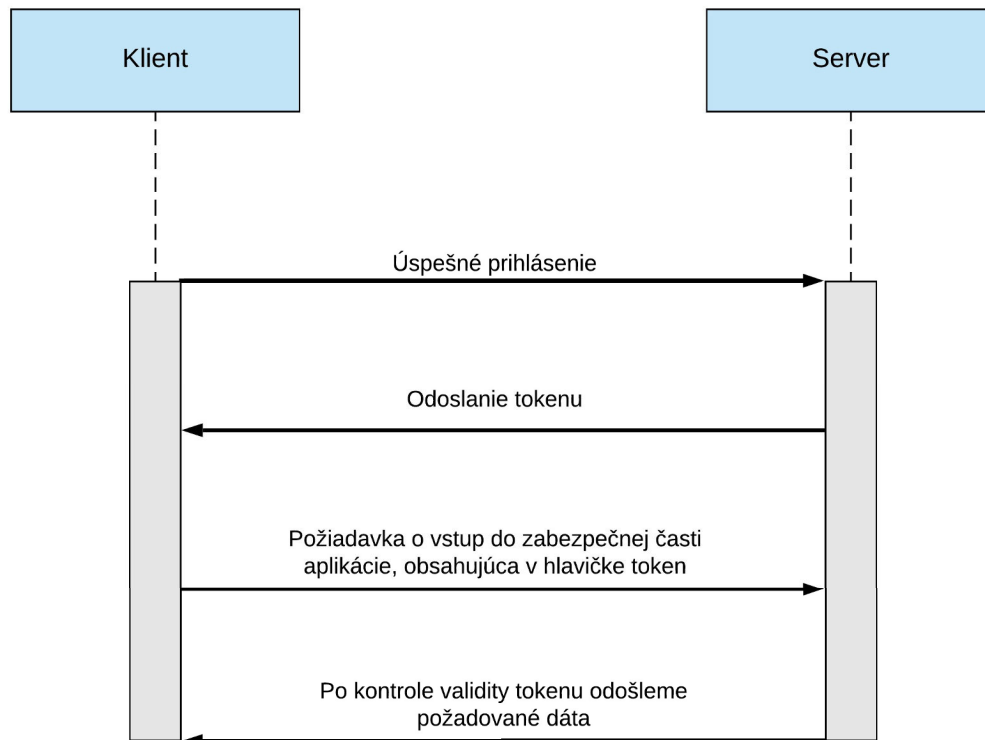
2.9.2 Autentifikácia a autorizácia

Na základe špecifikácie požiadaviek a vykonanej analýzy, bude autentifikácia pre rolu organizátor a rolu vedúci sekcie rôzna. Organizátor musí zadať správne prihlasovacie meno a heslo, vedúci sekcie prihlasovací kód. Prihlasovací formulár sa odošle na server metódou POST. Príslušná funkcia zadané heslo zašifruje pomocou spoľahlivého Bcrypt algoritmu a porovná so zašifrovaným heslom v databáze. Ak sú zhodné, pre používateľa sa vytvorí prihlasovací token. Pri každom prihlásení sa vytvára nový token, čo umožňuje byť prihlásený na viacerých zariadeniach súčasne. Zakódovaný token sa skladá z troch častí. Prvá časť je zakódované id používateľa, druhá časť je zakódovaná časová známka vzniku a tretia časť je ľubovoľný reťazec, ktorý poskytne správca databázy alebo iná poverená osoba.

K jednotlivým stránkam aplikácie sa pristupuje pomocou express routing. To platí pre metódy GET aj POST. Všetky routes môžeme rozdeliť do dvoch kategórií: verejné a chránené. Zatiaľ čo verejné sú prístupné všetkým, pre prístup k chránenej route je nutné byť autorizovaný. Autorizáciu vykonávame pomocou middleware funkcie, ktorú pridáme ako argument ku všetkým

routes, ktoré chceme autorizovať. Táto middleware funkcia sa spustí ešte pred samotnou route funkciou. Ak používateľ nie je autorizovaný, je vyhodnená chyba a route funkcia sa nezavolá. [19]

Keďže organizátor a vedúci sekcie sa prihlasujú rozličnými spôsobmi, každý má vlastnú autorizačnú funkciu. To znamená, že routes určené pre organizátora vyžadujú autorizáciu organizátora a routes určené pre vedúceho sekcie autorizáciu vedúceho sekcie. Žiadna z rolí sa preto nedostane k častiam aplikácie určených pre inú rolu.



Obrázok č. 16: Sekvenčný diagram úspešného vstupu do zabezpečenej časti aplikácie

2.10 Klient

2.10.1 Vkladanie dynamického obsahu

Aplikácia pracuje s dynamickým obsahom. Dynamický obsah generujeme prostredníctvom knižnice EJS, ktorá pracuje so súbormi s koncovkou .ejs. Jedná sa o html

súbory, do ktorých sa ešte na servery vložia objekty s dynamickým obsahom. U klienta následne pracujeme s atribútmi vložených objektov a na základe týchto atribútov prispôbujeme vzhľad zobrazovanej webstránky. Niektoré časti webstránok sa v aplikácií vyskytujú opakovane. Tie umiestnime do priečinka `partials`. EJS nám umožňuje tieto časti dynamicky generovať na mieste, kde potrebujeme. Vyhneme sa tým duplicité kódu. Následne pri zmene dát v aplikácií už neaktualizujeme celú stránku, ale len príslušnú časť. To realizujeme systémom tried a atribútov, ktoré priradíme jednotlivým DOM elementom. Pomocou triedy alebo atribútu vyberieme príslušný element a vykonáme požadovanú operáciu.

2.10.2 Podpora naprieč prehliadačmi

V aplikácií potrebujeme zadávať čas. Problém je, že formulárové pole typu čas každý prehliadač implementuje odlišným spôsobom a implementácie sú vo väčšine prípadov veľmi nepraktické. Rozhodli sme sa preto využiť knižnicu `timepicker.js`, ktorá síce nepatrí k najpoužívanejším, no plne vyhovuje našim potrebám a funguje vo všetkých prehliadačoch. S touto knižnicou dokážeme zadať čas na dva kliky, zadávanie času podrobnejšie opíšeme v podkapitole 3.2 Používateľské rozhranie.

Aplikácia funguje v prehliadačoch Google Chrome, Mozilla Firefox, Microsoft Edge a Opera. Aplikácia nefunguje v Internet Explorer, ktorý ešte stále plne nepodporuje EcmaScript 6. Dosiahnúť funkčnosť aj v tomto prehliadači by sme mohli konverziou na staršiu verziu JavaScriptu pomocou technológie Babel. Aplikáciu by to však zväčšilo a keďže zadávateľ nepožaduje funkčnosť v tomto prehliadači, nebudeme to robiť.

2.10.3 Design aplikácie

Design aplikácie tvoríme pomocou CSS3. CSS3 umožňuje používanie premenných, do ktorých si uložíme informácie o farbách, prechodoch, type a veľkosti písma. Následne využívame tieto premenné pri vytváraní designu aplikácie. Výhodou je, že zmenou jedinej premennej dosiahneme zmenu vo všetkých častiach aplikácie, v ktorých je premenná použitá.[20] Design aplikácie nám pomôže vylepšiť už spomenutá knižnica `timepicker.js`.

2.10.4 Zariadenia s menšou šírkou obrazovky

Aplikácia je plne použiteľná aj na zariadeniach s menšou šírkou obrazovky. Vzhľadom na skutočnosť, že rozvrh môže obsahovať väčšie množstvo miestností, sekcií, príspevkov, úprava rozvrhu a príspevkov je menej praktická ako pri väčšej obrazovke. Prispôbenie malej obrazovky slúži najmä vedúcim sekcie a účastníkom, aby si pozreli rozvrh na mobile alebo tablete, ak nemajú k dispozícii zariadenie s väčšou obrazovkou.

V tejto kapitole sme vykonali bližšiu analýzu požiadaviek a následne vytvorili návrh systému na manažovanie konferencií. V nasledujúcej kapitole aplikáciu implementujeme podľa vytvoreného návrhu.

3 Implementácia

V tejto kapitole implementujeme navrhnutú aplikáciu. Implementácia tejto aplikácie vyžadovala vyriešenie veľkého množstva problémov, nie je možné ich vzhľadom na obmedzený rozsah práce opísať všetky. Popíšeme preto niektoré z najpodstatnejších častí implementácie.

3.1 Export a import modulov

V aplikácií využívame export a import modulov, ktorý umožňuje vyššiu modularitu aplikácie. Vytvárame jednotlivé moduly, z ktorých každý má definovanú úlohu. Keď niektorý modul potrebuje použiť funkcionality iného modulu, jednoducho si ho importuje. Jedným z modulov je modul `basicView`, ktorý udržiava objekty z DOM domény. Ostatné moduly tak môžu k DOM objektom pristupovať pomocou názvu, ktorý si v tomto module určíme.[21] Príklad takéhoto importu vidíme na obrázku č. 17.

```
views > js > controllers > JS conferenceInfoController.js > ...
1  import { elementsCreateConference } from "../views/basicView.js";
2  import * as conferenceView from "../views/conferenceView.js";
3
4  const socket = io();
5
6  //create conference
7
8  elementsCreateConference.$createForm.addEventListener("submit", () => {
9    event.preventDefault();
10   const input = conferenceView.getCreateFormInput();
```

```
views > js > views > JS basicView.js > ...
4  export const elementsCreateConference = {
5    $createConference: document.querySelector(".create-conference"),
6    $createForm: document.querySelector("#create-form"),
7    $inputConferenceName: document.querySelector("#conference-name"),
8    $inputConferenceStart: document.querySelector("#conference-start"),
9    $inputConferenceFinish: document.querySelector("#conference-finish"),
10   $conferenceTitle: document.querySelector(".conference-title"),
11  };
```

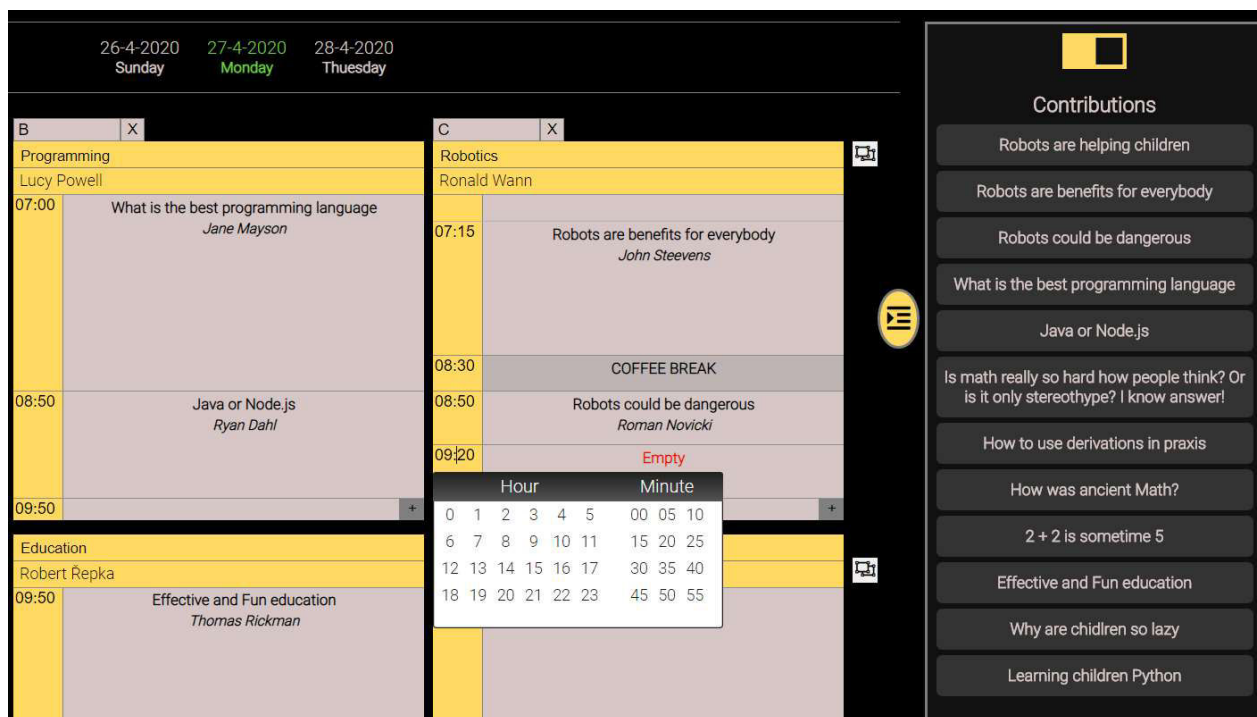
Obrázok č. 17: Import objektov z `basicView` modulu

Riešenie importu zvolíme aj pre funkcie, ktoré sa volajú z viacerých modulov. Takéto funkcie zoskupíme podľa logického hľadiska a pre každú skupinu vytvoríme modul. Týmto spôsobom dosiahneme zníženie duplicity kódu.

Takmer všetky moduly využívajú spoločný modul Feedback. Jeho úlohou je informovať používateľa, či jeho akcia bola úspešná alebo neúspešná. Informácia sa používateľovi zobrazí prostredníctvom malého okna v strede obrazovky. Používame ho najmä na miestach, kde na prvý pohľad nemusí byť jasné, či operácia bola úspešná. Feedback informuje používateľa aj o dôvode neúspechu akcie.

3.2 Používateľské rozhranie

Na obrázku č. 18 vidíme editáciu rozvrhu organizátorom. Pomocou žltého tlačidla pridávame nové miestnosti, pomocou bieleho tlačidla vytvoríme novú skupinu sekcií. Toto tlačidlo sa nachádza pri všetkých existujúcich skupinách sekcií a využíva šablónu. Použitím šablóny sa skopírujú prvky rozvrhu existujúcej skupiny sekcií, posunú sa o príslušný čas a následne sa vložia ako nová skupina sekcií.



Obrázok č. 18: Používateľské rozhranie organizátora – editácia rozvrhu

Organizátor si vytvára a upravuje prvky rozvrhu podľa svojej potreby. Pri tom musí zostať zachované rovnobežné zobrazenie príspevkov v rovnakých časoch. Problém čiastočne vyriešime tým, že veľkosť jednotlivých prvkov rozvrhu bude závisieť od dĺžky trvania príspevku, resp. prestávky. Môžu však nastať ďalšie problémy ako napríklad prípad, keď jednotlivé sekcie nezačínajú v rovnaký čas. Pre každú skupinu sekcií preto pri zmene prepočítavame globálne minimum. Na začiatok sekcií, ktoré začínajú neskôr ako je toto minimum, vložíme prázdny prvok príslušnej veľkosti.

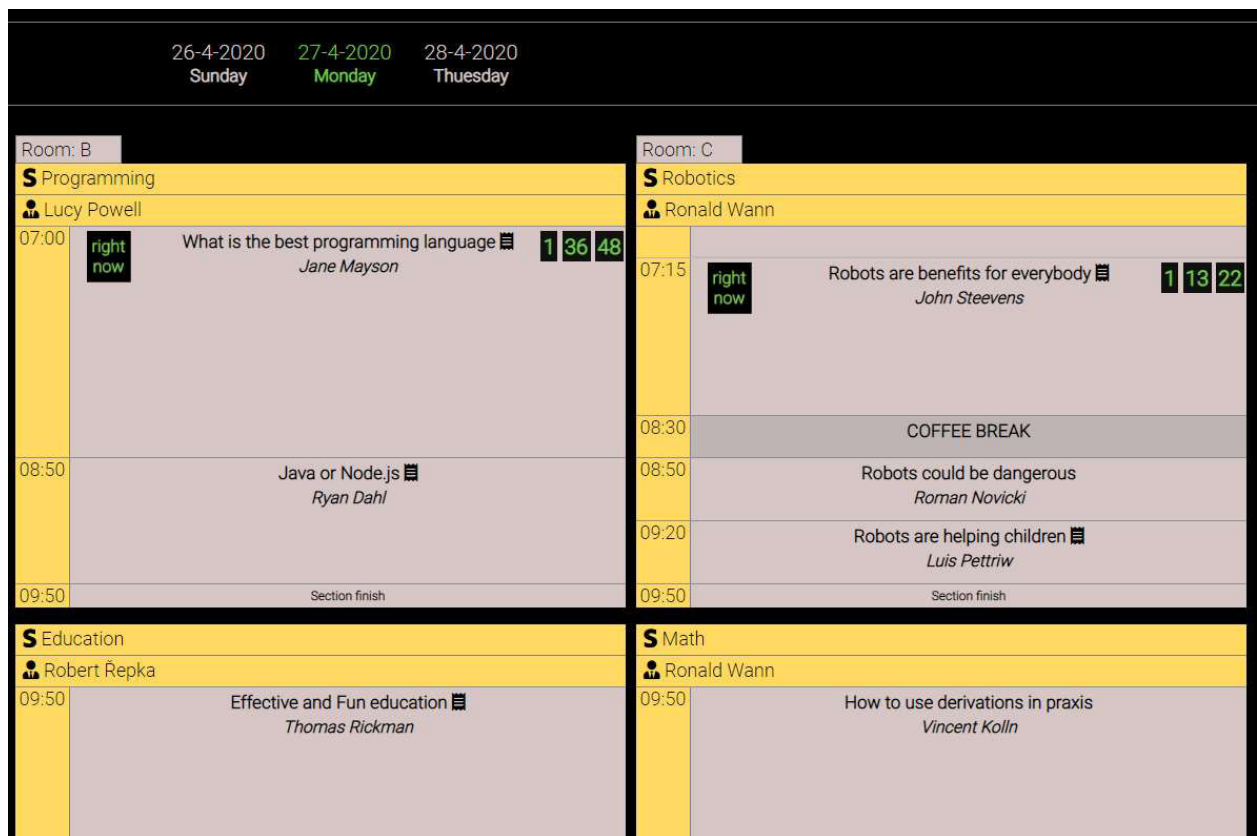
Vkladanie času je realizované prostredníctvom knižnice `timepicker.js`, ktorá už bola spomenutá v podkapitole 2.10.2. Túto knižnicu sme si prispôbili svojim potrebám. Vkladanie času je realizované prostredníctvom malého okna, ktoré sa zobrazí po kliknutí na textové pole. Používateľ môže do samotného textového poľa vložiť akýkoľvek text, my však potrebujeme, aby to bol korektný čas. To skontrolujeme pomocou regulárneho výrazu. Taktiež je nutné kontrolovať, či je zadaný čas korektný – nemôže sa stať, aby bol čas začiatku predchádzajúceho prvku rozvrhu vyšší alebo rovný ako zadávaný čas a rovnako čas nasledujúceho príspevku musí byť vyšší. V prípade vloženia nekorektného údaja, dané textové pole sčervená a ostatné textové polia rozvrhu sa zablokujú. Taktiež sa v tej chvíli znefunkčnia tlačidlá na pridávanie buniek rozvrhu, nových miestností a nových skupín sekcií. Cieľom je zachovanie rovnobežného zobrazenia príspevkov v rovnakých časoch a zachovanie konzistencie medzi rozvrhom u klienta a rozvrhom v databáze. Chybne zadaný čas sa na server nepošle a používateľ musí údaj opraviť na korektný, až následne dôjde k odblokovaniu tlačidiel a zmene času príspevku v databáze.

Treba dohliadnuť, aby sa časy jednotlivých skupín sekcií neprekrývali. V tomto prípade stačí kontrolovať prvý a posledný príspevok sekcie, pretože ostatné časy sú už porovnávané s časmi svojich susedných príspevkov a preto sú korektné. Ak zmeníme čas prvého alebo posledného príspevku sekcie na taký, ktorý sa prekrýva s predchádzajúcou, resp. nasledujúcou skupinou sekcií, opäť dôjde k začervneniu textového poľa a zablokovaniu ostatných polí. Keďže v tomto prípade nemusí byť na prvý pohľad jasné v čom je problém, zobrazí sa hláška o chybe s jej popisom.

Podobné problémy musíme riešiť aj pri pridávaní nových prvkov rozvrhu. Nový prvok rozvrhu má prednastavenú určitú časovú dĺžku, ktorú si používateľ vyberie v nastaveniach aplikácie. V prípade, že by nový prvok presahoval do ďalšej skupiny sekcií, jeho dĺžka sa

adekvátne zmenší. Ak už nie je žiadny priestor na jeho pridanie, teda daná sekcia končí presne v čase, keď začína nasledujúca skupina sekcií, k pridaniu prvku nedôjde a používateľ dostane chybovú hlášku. Obdobne vyriešime aj situáciu, keď by mal prvok rozvrhu presahovať do nasledujúceho dňa. Presah do nasledujúceho dňa by mohol nastať aj pridaním novej skupiny sekcií. Ak má dôjsť k takémuto presahu, opäť upozorníme používateľa na chybu.

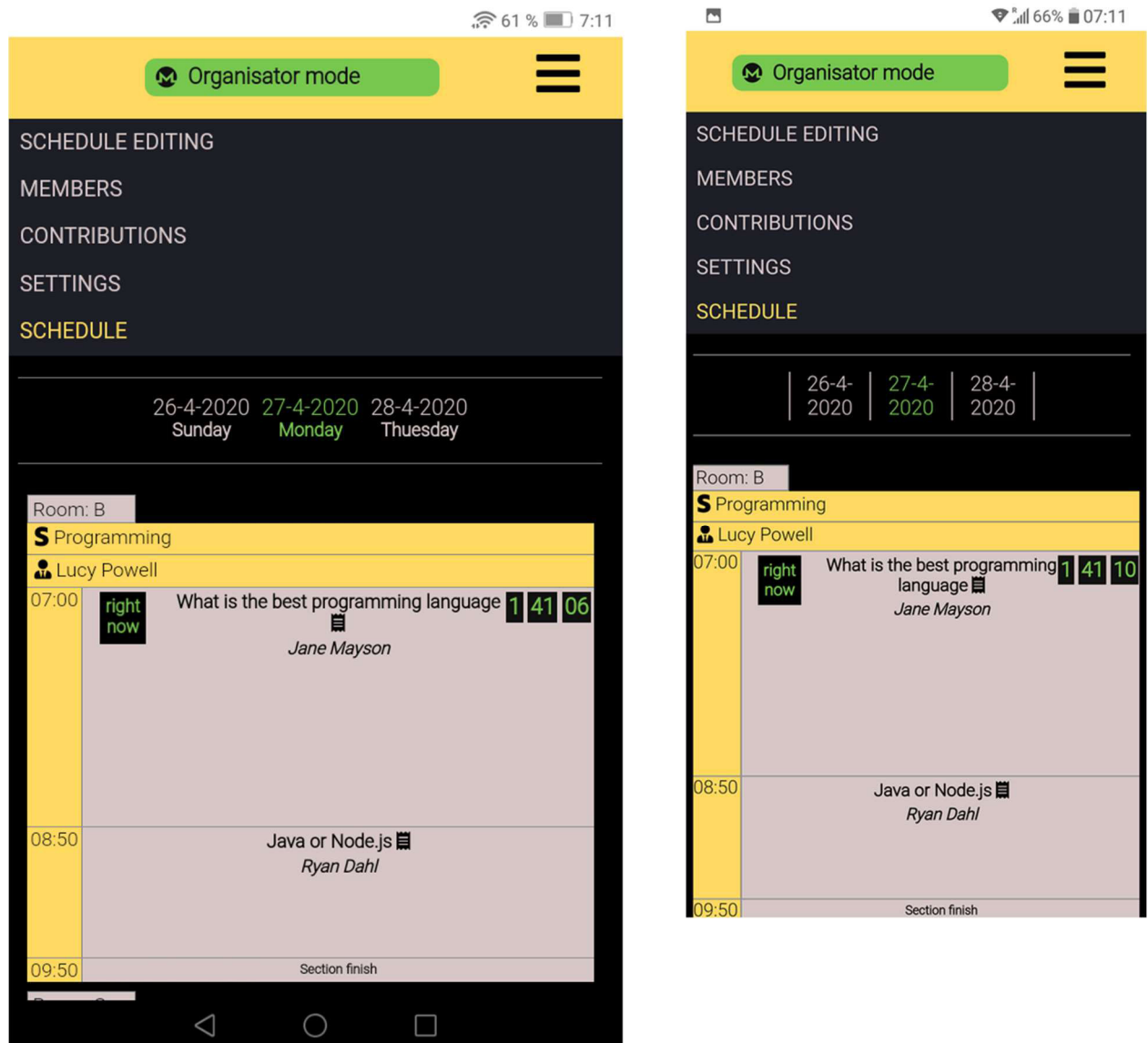
Konkrétne príspevky a vedúcich sekcie vkladáme do rozvrhu potáhovaním elementov z bočného panela. Pomocou čierneho tlačidla prepíname medzi príspevkami a vedúcimi sekcie. Následne potiahnutím umiestňujeme element na vybraný prvok rozvrhu. Dve zelené čiary signalizujú, nad ktorým prvkom rozvrhu sme. Následným pustením tlačidla myši prvok umiestnime. Využívame udalosti dragstart, dragover, dragleave, dragend a drop. Pri vkladaní musíme zabezpečiť, aby sme príspevky mohli vložiť iba na prvky rozvrhu a vedúcich sekcie iba na polia určené pre mená vedúcich sekcií. To zabezpečíme kontrolou atribútov potáhovaných prvkov. V prípade vloženia na nesprávne miesto, upozorníme používateľa na chybu.



Obrázok č. 19: Používateľské rozhranie návštevníka počas priebehu konferencie – rozvrh

Na obrázku č. 19 vidíme rozvrh z pohľadu návštevníka konferencie. Ak konferencia práve prebieha, návštevník vidí, ktoré príspevky práve prebiehajú a koľko času zostáva do ich konca. Kliknutím na názov príspevku alebo ikonu abstraktu sa zobrazí abstrakt príspevku a možnosť stiahnuť si článok.

3.3 Zobrazovanie na zariadeniach s menšou obrazovkou



Obrázok č. 20: Rozvrh z pohľadu organizátora na zariadeniach s menšou obrazovkou

Na obrázku č.20 vidíme zobrazenie rozvrhu počas priebehu konferencie z pohľadu organizátora na zariadeniach s malou šírkou obrazovky. Vľavo na menšom tablete so šírkou obrazovky 450 pixelov a vpravo na mobile so šírkou obrazovky 360 pixelov.

Pri implementácii aplikácie sme zvolili prístup mobile-first, čo znamená, že najskôr vytvoríme zobrazenie aplikácie na najmenšej obrazovke a následne so zväčšujúcou sa šírkou obrazovky upravujeme zobrazenie elementov podľa potreby. Šírke obrazovky sme prispôbili väčšinu elementov stránky. Názov konferencie v hlavičke sa zobrazí až od šírky obrazovky 600px. Navigácia pre obrazovky do 540px je vertikálna a je možné ju skryť pomocou tlačidla, čím používateľ získa väčšie množstvo priestoru pre rozvrh. Až od veľkosti 450px vidíme pod dátumom aj konkrétny deň v týždni. Väčšina elementov stránky sa veľkosťou aj zobrazením pri zmenách obrazovky priebežne prispôbuje. Pri zmene zobrazenia elementov, napríklad z vertikálneho na horizontálne, sa podľa potreby menia jednotlivé margins a paddings.[22]

3.4 Časovače

Časovač máme uložený na servery, pretože prístup k nemu potrebujú všetci klienti – potrebujú vidieť, koľko času zostáva do konca príspevku v konkrétnej miestnosti. Počas priebehu konferencie môže bežať simultánne niekoľko príspevkov, každý v jednej miestnosti. Bolo by neefektívne vytvárať pre každý bežiaci príspevok časovač. Naopak, vytvoríme jediný časovač.

Pôvodne sme časovač implementovali pomocou funkcie `SetInterval`, ktorá sa spúšťala každú sekundu a odpočítavala sekundu z času bežiacich príspevkov. Táto funkcia však nie je pre presné časovanie vhodná, nezaručuje presnosť – ovplyvňuje ju kvalita pripojenia medzi klientom a serverom. Napríklad u príspevku dlhého 10 minút, nepresnosť môže dosiahnuť aj 10 sekúnd, čo už nie je zanedbateľné.

Presné časovanie sa dá dosiahnuť pomocou dátového typu `datum`, kde môžeme interval vypočítať na základe času, ktorý ubehol od bodu štartu. Aj v tomto prípade však potrebujeme použiť na časovanie posielania správ klientovi nepresný `setInterval`. Napriek tomu, že teraz už je posielaný čas korektný, neposielame ho presne každú sekundu. Keď si zobrazíme prichádzajúci čas ku klientovi na viac desiatinných miest, vyzerá to asi ako nasledujúca postupnosť, udávaná v sekundách: 35.322, 34.214, 33.107, 32.004, 30.974. Pri posledných dvoch prvkoch postupnosti

nastáva problém. Hoci čas už je korektný, aktualizácie času neodosielame presne každú sekundu. Keďže časy zobrazené v aplikácii sú zaokrúhľované na celé sekundy, občas by sa stalo, že je jedna sekunda vynechaná, čo by nepôsobilo dobrým dojmom. Preto časovaču nastavíme vyššiu citlivosť, zvolíme aktualizáciu času každých 0.2 sekundy. Takáto zobrazovacia nepresnosť už nie je pre náš účel problém.[23]

3.5 Zobrazovanie v reálnom čase bez potreby aktualizácie stránky

Vybrané časti aplikácie naprogramujeme tak, aby sa zmeny vykonané jedným používateľom zobrazili v reálnom čase a bez potreby aktualizácie stránky u ostatných používateľov. Na obrázku č. 21 vidíme implementáciu Socket.io, konkrétne príklad mazania príspevku. Kliknutím na ikonu klient pošle serveru správu „deleteContribution“ aj s id príspevku. Server čaká na správu s týmto názvom. V momente ako takúto správu dostane, príspevok zmaže z databázy. Následne posiela dve správy späť. Prvá správa volaná na objekt io sa odošle všetkým klientom – je to informácia o zmazaní príspevku. Druhá správa volaná na objekt socket sa už odošle iba klientovi, ktorý vyslal pôvodnú správu. Ide o zobrazenie okna so správou o úspechu zmazania.

```
views > js > controllers > JS contributionController.js > ...
```

```
65 elementsContribution.$enrollmentItems.addEventListener("click", () => {
66   if (event.target.matches(".delete-icon, .delete-icon *")) {
67     feedbackView.removeFeedback();
68     const id = event.target.closest(".enrollment-item").dataset.id;
69     socket.emit("deleteContribution", id);
70     // also if (event.target.matches(".edit-icon, .edit-icon *")) {
```

```
server > JS index.js > io.on("connection") callback
```

```
216 socket.on("deleteContribution", async (id) => {
217   try {
218     await Contribution.findOneAndDelete({ _id: id });
219     io.emit("contributionDeleted", id);
220     socket.emit("showSuccess", {
221       text: "The contribution was successfully deleted",
222     });
223   } catch (e) {
```

Obrázok č. 21: Implementácia Socket.io

V tejto kapitole sme vykonali implementáciu aplikácie. Aby sme si boli istí, že všetko funguje podľa špecifikácie a analýzy požiadaviek, aplikáciu musíme otestovať. Testovanie vykonáme v nasledujúcej kapitole.

4 Testovanie

Čiastočné testovanie aplikácie bolo vykonávané už priebežne počas vývoja. Väčšie a komplexnejšie testovanie je potrebné vykonať nakoniec. Podľa pôvodného plánu sa aplikácia mala najskôr otestovať pomocou testovacích scenárov a následne na niektorej zo skutočných konferencií prebiehajúcich na FMFI. Vzhľadom na pandémiu COVID-19 a mimoriadnu situáciu v dobe písania práce, testovanie na konferencií nie je možné vykonať. Vykonáme preto testovanie podľa testovacích scenárov a následne dáme aplikáciu otestovať niekoľkým osobám, medzi ktorými budú zastúpené osoby z oblasti informatiky aj mimo nej. Pripomienky zapracujeme do aplikácie. Testovanie podľa testovacích scenárov sme rozdelili do štyroch skupín.[24]

4.1 Testovacie scenáre pre manažovanie príspevkov a používateľských účtov

- **Manažovanie účtov organizátorov:** Prihlásime sa na účet hlavného organizátora. Vytvoríme 3 ďalšie účty organizátorov. Následne upravíme ich prihlasovacie mená. Prihlásime sa na účet jedného z práve vytvorených organizátorov v inom prehliadači. Kontrola, že vytvorený organizátor nemá prístup k upravovaniu účtov ostatných organizátorov. Z účtu hlavného organizátora zmažeme prístup práve prihlásenému organizátorovi. Ten je automaticky odhlásený a opätovné prihlásenie nie je možné.
- **Manažovanie účtov vedúcich sekcie:** Tento testovací scenár vykonáme z účtu hlavného a iného organizátora. Vytvoríme vedúceho sekcie bez vloženia prihlasovacieho kódu. Následne vytvoríme iného vedúceho sekcie, kde vložíme aj prihlasovací kód. Otestujeme jeho prihlásenie. Najskôr vložíme nesprávny kód, potom správny. Zmeníme mu prihlasovací kód, následne otestujeme prihlásenie aj s týmto novým prihlasovacím kódom. Zmažeme vedúceho sekcie.
- **Manažovanie príspevkov:** Tento testovací scenár vykonáme z účtu hlavného a iného organizátora. Vytvoríme niekoľko príspevkov. Niektorým z nich pridáme aj článok, čo je

súbor vo formáte PDF. Pokúsime sa vložiť aj súbor v inom formáte, čo je zakázané. Následne príspevky upravujeme a niektoré z nich zmažeme.

- **Manažovanie – viac organizátorov súčasne:** Jednotlivé body predchádzajúcich troch testovacích scenárov vykonávame prostredníctvom rôznych účtov organizátorov prihlásených na rôznych zariadeniach, v rôznych prehliadačoch. Otestujeme zobrazovanie dát v reálnom čase a ich konzistenciu.

4.2 Testovacie scenáre pre tvorbu rozvrhu

- **Malá konferencia – 1 deň, 1 miestnosť, 1 sekcia:** Vytvoríme miestnosť A. V tejto miestnosti sa koná sekcia, ktorú si ľubovoľne pomenujeme. Sekcií priradíme vedúceho. Vytvoríme tri prvky rozvrhu v časoch od 8:00 do 10:00, od 10:00 do 10:30 a od 10:30 do 11:00. Prvkom rozvrhu priradíme príspevky.
- **Malá konferencia s prestávkou:** Vykonáme predchádzajúci testovací scenár a medzi príspevky vložíme prestávku v trvaní od 10:30 do 10:40, dĺžku posledného príspevku skrátime na 20 minút. Kontrolujeme odlišné sfarbenie prestávky. Následne na miesto prestávky vložíme opäť bežný príspevok, kontrolujeme zmenu sfarbenia.
- **Malá konferencia s dvomi sekciami:** Vykonáme predchádzajúci testovací scenár s tým, že od 11:00 začne nová sekcia. Tú vložíme prostredníctvom šablóny. Počet aj dĺžka jej príspevkov sú rovnaké ako u prvej sekcie.
- **Malá konferencia s dvomi sekciami a obedovou prestávkou:** vykonáme predchádzajúci scenár s tým, že medzi sekciami vložíme ešte obedovú prestávku v trvaní 45 minút.
- **Malá konferencia s tromi miestnosťami a šiestimi sekciami:** vykonáme predchádzajúci scenár a následne vložíme dve nové miestnosti. Týmto miestnostiam priradíme vedúcich sekcie a príspevky. Následne jednu z miestností zmažeme.
- **Malá konferencia – zarovnanie časov:** vykonáme predchádzajúci scenár a následne ľubovoľne meníme časy jednotlivých príspevkov. Kontrolujeme, či sú miestnosti za každých okolností zarovnané podľa časov. Zmeníme aj časy prvých príspevkov miestností.

- **Malá konferencia – vkladanie nekorektných časov:** vykonáme predchádzajúci scenár a následne meníme časy príspevkov na nekorektné. Kontrolujeme, že nekorektné časy neboli vložené do systému.
- **Malá konferencia – vkladanie nových prvkov rozvrhu a skupín sekcií:** pokiaľ je to možné, vkladáme nové prvky rozvrhu. Kontrolujeme, že systém nám dovolí vkladať nové prvky iba do začiatku nasledujúcej sekcie. V prípade, že nami nastavená dĺžka vkladaneho prvku rozvrhu je menšia ako čas zostávajúci do začiatku nasledujúcej skupiny sekcií, veľkosť vkladaneho prvku rozvrhu je adekvátne znížená. Ak už za sekciou nenasleduje ďalšia sekcia, systém dovolí príspevky vkladať maximálne do polnoci. Rovnaký test vykonáme aj s vkladáním nových skupín sekcií.
- **Malá konferencia – zmena poradia príspevkov:** príspevkom jednotlivých sekcií meníme poradie, kontrolujeme, či sa časy zmenili korektne a dĺžka príspevkov ostala zachovaná
- **Malá konferencia – mazanie príspevkov:** z rozvrhu zmažeme niekoľko príspevkov. Kontrolujeme či sa časy ostatných príspevkov posunuli korektne.
- **Veľká konferencia:** vytvoríme niekoľkodňovú konferenciu, následne pre niektoré z dní opakujeme testovacie scenáre pre malú konferenciu
- **Zmena nastavení:** zmeníme niektoré nastavenia ako názov konferencie, dátum konferencie, predvolenú dĺžku príspevkov konferencie alebo predvolenú veľkosť prvkov rozvrhu. Následne opakujeme testovací scenár Veľká konferencia so zmenenými nastaveniami.
- **Konferencia s veľmi krátkymi príspevkami:** Prvkom rozvrhu dáme veľmi malú dĺžku trvania, napr. 10 minút. Následne v nastaveniach upravujeme výšku prvkov rozvrhu. Kontrolujeme korektnosť zmien.
- **Úprava rozvrhu viacerými organizátormi:** Vykonáme scenár veľká konferencia prostredníctvom rôznych účtov organizátorov prihlásených na rôznych zariadeniach, v rôznych prehliadačoch. Otestujeme zobrazovanie dát v reálnom čase a ich konzistenciu.
- **Zhoda rozvrhov organizátora a návštevníka:** skontrolujeme, či je rozvrh vytvorený organizátorom zhodný s rozvrhom zobrazovaným návštevníkovi.

- **Zobrazenie abstraktov a článkov návštevníkom:** skontrolujeme, či sa po kliknutí na názov príspevku alebo ikonu abstraktu zobrazí abstrakt – za predpokladu, že je k dispozícii. Taktiež skontrolujeme, či si návštevník môže stiahnuť článok.

4.3 Testovacie scenáre pre priebeh konferencie v reálnom čase

Program kontroluje aktuálny dátum a na základe toho povoľuje niektoré činnosti iba v daný deň. Napríklad vedúci sekcie môže spúšťať príspevky iba v deň, keď sú plánované. Pri testovaní konferencie v reálnom čase je potrebné na túto skutočnosť pamätať a dátum prispôbiť aktuálnemu dátumu. Druhou možnosťou je v kóde nasimulovať aktuálny deň.

- **Vedúci sekcie - prihlásenie:** vytvoríme konferenciu podľa testovacieho scenára Veľká konferencia. Následne jednému z vedúcich sekcií priradíme niekoľko sekcií vo viacerých dňoch konferencie. Skontrolujeme, či vedúci po prihlásení vidí len svoje sekcie.
- **Vedúci sekcie - zmena poradia príspevkov:** príspevkom meníme poradie, kontrolujeme či sa časy zmenili korektne a dĺžka príspevkov ostala zachovaná
- **Vedúci sekcie - zrušenie príspevku:** zrušíme niektoré príspevky, následne niektoré z nich zase aktivujeme.
- **Vedúci sekcie - spustenie/ukončenie príspevku:** spustíme niektorý z príspevkov. Kontrolujeme, že nie je možné spustiť iné príspevky danej sekcie. Necháme príspevok dobehnúť až dokonca. Skontrolujeme, či časovač príspevku zostáva zobrazený s hodnotou 0:00:00, až pokiaľ príspevok neukončí vedúci sekcie.
- **Vedúci sekcie – odhlásenie:** po spustení príspevku sa odhlásime z účtu. Následne sa opäť prihlásime. Kontrolujeme, že príspevok beží s korektným zostávajúcim časom
- **Vedúci sekcie - viacero príspevkov bežiacich súčasne:** viacero vedúcich sekcie spustí príspevky. Následne niektoré z nich ukončíme. Sledujeme korektnosť zmien a to, že časovače príspevkov, ktoré sme neukončili, bežia naďalej.
- **Kontrola zobrazenia v reálnom čase:** kontrolujeme, či sa zmeny vykonané vedúcim sekcie premietli do rozvrhu účastníka bez potreby aktualizácie stránky.
- **Časovač - prihlásenie:** prihlásime sa na časovač. Skontrolujeme, či vidíme iba sekcie prebiehajúce v daný deň. Vyberieme jednu zo sekcií, ktorá práve prebieha. Skontrolujeme, či je čas na časovači korektný a zhodný s časom v rozvrhu

- **Časovač zmena farby:** skontrolujeme, či pri posledných 5 minútach dôjde k zmene farby časovača
- **Časovač korektnosť plynutia času:** kontrolujeme, či je uplynutý čas na časovači rovnaký ako uplynutý čas v skutočnosti

4.4 Testovanie zabezpečenia aplikácie

Aplikáciu otestujeme aj z bezpečnostného hľadiska. Nie je možné sa dostať bez prihlásenia do častí aplikácie, ktoré sú určené len prihlásenému používateľovi. Taktiež nie je možné sa dostať do častí aplikácie, ktoré sú určené pre inú rolu. Nie je to možné ani v prípade, že by osoba poznala presnú URL adresu.

V tejto kapitole sme aplikáciu otestovali, čím sme sa presvedčili, že implementácia aplikácie bola úspešná.

Záver

Cieľom práce bolo vytvoriť real-time webovú aplikáciu na podporu priebehu konferencie. Analýza požiadaviek nám pomohla vytvoriť vhodný návrh, podľa ktorého sme následne aplikáciu implementovali, čo viedlo k naplneniu cieľa práce.

V prvej kapitole sme si popísali priebeh konferencie, problémy, ktoré momentálne pri organizácii konferencií sú a požiadavky zadávateľa na aplikáciu. Taktiež sme vykonali analýzu podobného systému. Následne v druhej kapitole sme podrobnejšie analyzovali požiadavky a na základe analýzy vytvorili vhodný návrh aplikácie. Vybrali sme architektúru frontend-backend s podporou Socket.io, ktorá nám umožnila vykonávať zmeny u všetkých používateľov aplikácie bez potreby aktualizácie stránky. Dáta si ukladáme v MongoDB databáze a čiastočne na servery. Pri návrhu systému sme postupovali podľa návrhového vzoru MVC. V tretej kapitole sme navrhnutú aplikáciu implementovali. Postupovali sme takým spôsobom, aby bolo jej využitie možné aj na zariadeniach s menšou šírkou obrazovky. V štvrtej kapitole sme si vytvorili testovacie scenáre, pomocou ktorých sme otestovali, že aplikácia funguje podľa špecifikácie požiadaviek.

Pomocou webovej aplikácie sa nám podarilo vyriešiť problémy organizátorov konferencie a zároveň pridať niektoré prvky, ktoré umocnia zážitok návštevníkov. Rozvrh konferencie bude teraz obsahovať vždy správne a najaktuálnejšie údaje. Ak organizátor vloží do rozvrhu nekorektný údaj, program ho na chybu upozorní. Implementovali sme algoritmus, vďaka ktorému sú príspevky, ktoré bežia v rovnakom čase, vždy rovnobežne zarovnané. Tým sa sprehládni situácia pre návštevníka a zjednoduší výber príspevkov, ktoré chce navštíviť. Účastník konferencie bude mať v rozvrhu špeciálne zvýraznené práve prebiehajúce príspevky a tiež vyznačený čas zostávajúci do skončenia príspevku. To mu umožní lepšie si naplánovať, ktorých príspevkov sa chce zúčastniť, keďže niektoré môžu oproti pôvodnému plánu meškať. Navyše si môže prezerať abstrakty alebo stiahnuť články k jednotlivým príspevkom. Prínosom je aj zobrazenie zostávajúceho času pre prednášajúceho na vedľajšom monitore alebo tablete. Dokáže tak lepšie prispôbiť dĺžku svojej prednášky.

Rád by som ešte vyzdvihol prínosy práce pre mňa ako autora. Aplikáciu nebolo jednoduché naprogramovať. Keď som začínal, v programovacom jazyku JavaScript som sa cítil pomerne neisto. Počas vývoja som musel vyriešiť obrovské množstvo problémov, aby nakoniec všetko fungovalo tak, ako bolo vyžadované. Práve vďaka tomu som sa veľa vecí naučil. JavaScript sa stal mojim obľúbeným programovacím jazykom a mám pocit, že už by som v ňom dokázal vyriešiť akýkoľvek problém a naprogramovať akúkoľvek webovú aplikáciu.

Na úplný záver uvediem návrh na prípadné ďalšie vylepšenie aplikácie v budúcnosti. Účastník konferencie by si mohol vytvoriť vlastný plán príspevkov, ktorých sa chce zúčastniť. Následne by vznikol jeho vlastný rozvrh, kde by mal prehľadne vyznačené len svoje príspevky. Ak by došlo k zmene v jeho rozvrhu, prišlo by mu upozornenie o tejto zmene. Upozornenie by obsahovalo presnú informáciu, aká zmena nastala.

Zdroje

- [1] EVENUES, What is a Conference? eVenues. [online]. [cit. 10-05-2020]. Dostupné na internete: <https://evenues.com/event-planning-guide/what-is-a-conference>
- [2] MINNICK J., Responsive Web Design with HTML5 & CSS3. 8th edition, Cengage Learning, 2016.
- [3] MEETINGHAND, The Most Efficient & Easy-to-use Online Event Management Software. Meetinghand, 2020. [online]. [cit. 12-02-2020]. Dostupné na internete: meetinghand.com/
- [4] REJNKOVÁ P., Diagram aktivit. [online]. [cit. 28-03-2020]. Dostupné na internete: http://uml.czweb.org/diagram_aktivit.htm
- [5] TAGLIAFERRI L., How To Work with JSON in JavaScript. DigitalOcean. [online]. [cit. 13-02-2020]. Dostupné na internete: <https://www.digitalocean.com/community/tutorials/how-to-work-with-json-in-javascript>
- [6] YOUNG A., HARTER M., Node.js in practice. Manning, 2015.
- [7] NODES.JS – EXPRESS FRAMEWORK, Tutorialspoint. [online]. [cit. 14-02-2020]. Dostupné na internete: https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
- [8] SOCKET.IO, documentation. [online]. [cit. 14-02-2020]. Dostupné na internete: <https://socket.io/docs/>
- [9] BOTERHOVEN D., Why you should use BCrypt to hash passwords, Medium. [online]. [cit. 14-02-2020]. Dostupné na internete: <https://medium.com/@danboterhoven/why-you-should-use-bcrypt-to-hash-passwords-af330100b861>
- [10] JSON WEB TOKEN, documentation. [online]. [cit. 14-02-2020]. Dostupné na internete: <https://jwt.io/>
- [11] EJS, documentation. [online]. [cit. 12-03-2020]. Dostupné na internete: <https://ejs.co/>
- [12] MONGOOSE, documentation. [online]. [cit. 15-02-2020]. Dostupné na internete: <https://mongoosejs.com/>

- [13] HOWS D., MEMBREY P., PLUGGE E., MongoDB Basics. Apress. 2015.
- [14] SATHEESH M., D’MELLO B. J., KROL J., Web development with MongoDB and NodeJS, 2nd edition. Packt Publishing, 2015.
- [15] BECK W., App Architecture – Understanding Frontend, Backend and Web Servers. City Kitty Design. [online]. [cit. 22-04-2020]. Dostupné na ineternete: <http://citykittydesign.com/app-architecture-understanding-frontend-backend-and-web-servers/>
- [16] CHROME BROWSER, MVC Architecture. [online]. [cit. 14-03-2020]. Dostupné na internete: https://developer.chrome.com/apps/app_frameworks
- [17] TIMMS S., Mastering JavaScript Design Patterns. Packt Publishing, 2014.
- [18] JAVASCRIPT, documentation. [online]. [cit. 25-04-2020]. Dostupné na internete: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Async_await
- [19] QUINN S., You don’t need passport.js – Guide to node.js authentication. [online]. [cit. 26-04-2020]. Dostupné na internete: <https://softwareontheroad.com/nodejs-jwt-authentication-oauth/>
- [20] OHANS E., Everything you need to know about CSS variables. Freecodecamp. [online]. [cit. 27-04-2020]. Dostupné na internete: <https://www.freecodecamp.org/news/everything-you-need-to-know-about-css-variables-c74d922ea855/>
- [21] HAVERBEKE M., Eloquent Javascript. No Starch Press, 2018.
- [22] ENFROY A., Why mobile first design is the only 2019 strategy that will work. Searchenginewatch. [online]. [cit. 02-05-2020]. Dostupné na internete: <https://www.searchenginewatch.com/2019/03/05/mobile-first-design-for-2019/>
- [23] EDWARDS J., Creating accurate timers in JavaScript. Sitepoint. [online]. [cit. 04-05-2020]. Dostupné na internete: <https://www.sitepoint.com/creating-accurate-timers-in-javascript/>
- [24] 180+ WEB APPLICATION TESTING EXAMPLE TEST CASES. Software Testing Help. [online]. [cit. 04-05-2020]. Dostupné na internete: <https://www.softwaretestinghelp.com/sample-test-cases-testing-web-desktop-applications/>

Zoznam obrázkov

Obrázok č. 1: MeetingHand – Vytváranie udalosti

Obrázok č. 2: MeetingHand – Prehľad prihláseného organizátora

Obrázok č. 3: MeetingHand - Zmena bočného panela organizátora

Obrázok č. 4: MeetingHand - Vzhľad rozvrhu z pohľadu organizátora

Obrázok č. 5: MeetingHand - Vytváranie položky rozvrhu a možnosť voľby času vo formulári

Obrázok č. 6: MeetingHand - Stránka z pohľadu návštevníka

Obrázok č. 7: MeetingHand - Rozvrh z pohľadu návštevníka

Obrázok č. 8: Use Case diagram

Obrázok č. 9: Náskres tvorby rozvrhu konferencie

Obrázok č. 10: Zarovnanie rôzne dlhých príspevkov v rovnaký deň

Obrázok č. 11: Diagram aktivít – aktivity vedúceho sekcie počas konferencie

Obrázok č. 12: On-line manažovanie – vedúci sekcie

Obrázok č. 13: Diagram tried

Obrázok č. 14: Architektúra systému

Obrázok č. 15: Použitie návrhového vzoru MVC v aplikácií

Obrázok č. 16: Sekvenčný diagram úspešného vstupu do zabezpečenej časti aplikácie

Obrázok č. 17: Import objektov z basicView modulu

Obrázok č. 18: Používateľské rozhranie organizátora – editácia rozvrhu

Obrázok č. 19: Používateľské rozhranie návštevníka počas priebehu konferencie – rozvrh

Obrázok č. 20: Rozvrh z pohľadu organizátora na zariadeniach s menšou obrazovkou

Obrázok č. 21: Implementácia Socket.io

Zoznam príloh na CD

Zdrojový kód aplikácie

Manuál k spusteniu a údržbe systému na manažovanie konferencie v reálnom čase